Learning Preference Models:
A Marriage between Decision Theory and Machine Learning

par Margot Herin

Thèse de doctorat en informatique
dirigée par Patrice Perny et Nataliya Sokolovska
présentée et soutenue publiquement le 20/06/2025.

Jury :

Yann Chevaleyre	Rapporteur	Professeur, Université Paris Dauphine PSL
Michel Grabisch	Rapporteur	Professeur, Université Paris I Panthéon-Sorbonne
Isabelle Bloch	Examinatrice	Professeure, Sorbonne Université
Sébastien Destercke	Examinateur	Directeur de recherche CNRS, UTCompiègne
Eyke Hüllermeier	Examinateur	Professeur, Ludwig-Maximilians-Univ. München
Christophe Labreuche	Examinateur	Ingénieur R&D, Thales Recherche et Technologie
Patrice Perny	Directeur de thèse	Professeur, Sorbonne Université
Nataliya Sokolovska	Co-directrice de thèse	Professeure, Sorbonne Université

Remerciements

Tout d'abord, je souhaite exprimer ma plus profonde reconnaissance envers mon directeur de thèse, Patrice Perny, pour son investissement et son soutien sans faille, dont j'ai eu la chance inouïe de bénéficier. Cette généreuse disponibilité humaine, complétée par un humour des plus appréciables, sa vaste et profonde maîtrise du domaine de la théorie de la décision, ainsi que son excellence académique en général, font de lui, je crois, un encadrant de thèse d'exception. De la même manière, je remercie très sincèrement ma co-directrice de thèse, Nataliya Sokolovska, pour son écoute, ses attentions et ses conseils avisés, témoignants de sa large expertise en apprentissage statistique.

Ensuite, je souhaite remercier chaleureusement Michel Grabisch pour m'avoir fait l'honneur d'accepter d'être rapporteur de ce manuscrit. Ses travaux en théorie de la décision sont tout simplement immenses, et ont été un guide tout au long de ma thèse. Je remercie tout aussi chaleureusement Yann Chevaleyre, dont l'expertise colossale, à la fois en théorie de la décision et en apprentissage statistique, a donné lieu à des conseils précieux, et constitue pour moi une grande source d'inspiration.

Naturellement, je remercie Isabelle Bloch pour avoir accepté la présidence de ce jury, ainsi que pour ses attentions concernants le bon déroulement de ma thèse. Ensuite, je remercie Sébastien Destercke d'avoir accepté le rôle d'examinateur, mais aussi pour les remarques stimulantes qu'il a pu formuler lors de rencontres en conférence. Enfin, je remercie chaleureusement les examinateurs Eyke Hüllermeier et Christophe Labreuche, dont les travaux à la croisée de l'apprentissage statistique et de la théorie de la décision, ont constitué une référence pour moi.

Bien sûr, je tiens aussi à remercier l'équipe décision et RO du LIP6 pour l'accueil en 26-00, avec des mentions spéciales pour Thibaut Lust, collègue d'enseignement mais aussi ami et joyeux camarade du buisson ardent, Pierre-Henri Wuillemin, éternel voisin de palier et générateur de doctorants sympathiques, et Olivier Spanjaard, avec qui il est toujours agréable de discuter. Évidemment, une part très importante de mes remerciements est dédiée au clan 401, avec en premier lieu l'ancêtre qui m'a recueilli, Marvin, sans oublier l'ancêtre numéro 2, Gaspard, et puis bien sûr Clara et Mahdi à qui j'aurais presque envie de dédier cette thèse, tellement leur présence quotidienne et leur bavardages incessants ont compté pour moi. Enfin, merci à Dounia, la petite dernière, qui a su rendre ma fin de thèse un peu plus facile à coup d'opinions non conventionnelles et de fous rires tardifs. Je pense également à d'autres chouettes doctorants du LIP6 que sont Mohammed, Garance, Maxime, Jordan, avec une pensée spéciale pour Yann et Manuel, avec qui j'ai pu partager les joies de l'ATER. J'en profite pour remercier également mes supers encadrants de stage, Bertrand, Vincent et Marouane, qui m'ont lancé dans le monde de la recherche, et m'ont toujours apporté du soutien de loin.

Je souhaite maintenant remercier du fond du coeur mes amis de longue date, qui ont su apporter du soleil en tout temps, avec une place d'or pour mes deux colocataires et meilleures amies, Louise et Estelle, dont la présence a été d'un soutien incommensurable. Je pense ensuite bien sûr à mon meilleur ami Louis, mais aussi à mes meilleurs amis de toujours, le XBAR, j'ai nommé, Sonia, Gaby, Léo, Lucie, Flavia, Chloé, Juliette, Maks, Sacha, Maxime, Brams, Adam, Hector, Kevan, Ben, aux merveilleux copains de l'ENSTA, aux copines de Rio, avec mention spéciale pour Margaux qui a toujours su me transmettre du positif, pour finir avec un clin d'oeil à la Church, si accueillante.

Merci à toute ma famille qui m'est infiniment précieuse, en particulier à mes parents, qui ont su me donner tout ce qui compte, à mes quatre grands-parents d'exception, et à mon rayon de soleil de frère.

Finalement, merci à Jo dont je n'ai pas les mots pour décrire l'immensité du soutien et de la patience dont il a fait preuve durant ces années de thèse, et l'ampleur de l'amour que je lui porte.

Contents

List of Figures

List of Tables

Introduction

Moving forward is about making decisions. As individuals or as a group, we are frequently confronted with situations that demand evaluating and organizing the options at hand, and ultimately making a choice, or ranking or classifying these alternatives. The inherent complexity of these decisions often lies in the need to consider diverse viewpoints on the alternatives. For instance, the evaluation of development projects or public policies (e.g., climate policies [Doukas and Nikas, 2020], sustainable energy planning [Kumar et al., 2017, Pohl and Geldermann, 2024], energy supply resilience management [Siskos and Burgherr, 2022]) requires taking into account impacts of different kinds (e.g., environmental, economic, social) and/or the opinions of various experts and stakeholders. Similarly, on an individual level, a typical example is the choice of a travel route, which requires balancing multiple factors like cost, duration, carbon emissions, and comfort. Therefore, decision support systems, as well as systems capable of making automated decisions, play a crucial role in today's world – a key example being recommender systems. Yet, the effectiveness of these tools in meeting the needs of *decision-makers* depends on their ability to guide or act in accordance with *their preferences*, thus ensuring that the decisions made align with their values.

In this regard, a vast body of literature within the field of *decision theory*, which ranges from economics [Von Neumann and Morgenstern, 1944] and mathematical psychology [Tversky and Kahneman, 1981] to artificial intelligence [Bacchus and Grove, 1995, Boutilier et al., 1999], has introduced models capable of capturing decision-makers' preferences, often through parameters adjustable to their value systems. Among the different types of models, this thesis focuses value functions that assign to alternatives overall scores reflecting their attractiveness to the decision-maker. Examples of value function models include, the weighted sum, the ordered weighted average (OWA) [Yager, 1988], the weighted OWA [Torra, 1997], the Choquet integral [Grabisch, 1996], the Sugeno integral [Sugeno, 1974], the weighted Chebyshev norm [Wierzbicki, 1986], the expected utility [von Neumann and Morgenstern, 1947], the Choquet expected utility [Schmeidler, 1989], the multilinear utility [Keeney and Raiffa, 1976] and GAI-decomposable utility functions [Fishburn, 1970].

These models, while satisfying *normative* properties that ensure decision consistency, often exhibit strong *descriptive* capabilities, and can therefore account for complex decision-making behaviors. In particular, some models stand out for their ability to model *interactions* between viewpoints on the alternatives. For instance, models based on the Choquet integral, the multilinear utility, or the Sugeno integral, use a weighting system, called *capacity*, that assigns importance weights to each viewpoint, but also to all possible groups of viewpoints. Another relevant example is GAI-decomposable utility functions, which take the form of an additive decomposition in multivariate terms, capable of encoding interactions across all possible groups of viewpoints. However, due to the combinatorial nature of the possible interactions, the descriptive and normative power of these model is made possible by using a large number of parameters and constraints, that often grows exponentially with the number of viewpoints considered. As a result, methods for calibrating these parameters to the value system of a decision-maker (also known as *preference elicitation*) usually rely on prior restrictions of the model flexibility (e.g., by predefining the possible interacting groups of viewpoints) [Braziunas and Boutilier, 2005, Grabisch et al., 2008, Pelegrina et al., 2020a, Beliakov and Wu, 2021, Grabisch et al., 2022]. Thus, fully leveraging the descriptive richness of these models for problems of significant size (involving more than a dozen viewpoints, i.e., more than several thousand possible interactions) remains a challenge.

On the other hand, methods for adjusting model parameters based on examples have been developed in the field of *machine learning*, notably through *regularized empirical risk minimization* [Vapnik, 1995], which aims to identify the model that minimizes both the error on the examples and a regularization term to control the complexity of the learned model. When the model involves a large number of parameters (understood here as more than ten thousand), it is often desirable to control the model complexity by seeking a *sparse* model, where only a limited number of parameters are non-zero. This not only simplifies the model, making it easier to interpret, but also significantly reduces computational and memory costs during both storage and prediction phases. A standard approach to achieve this consists in employing *sparsity-inducing regularization* functions. These regularization functions—of which the ℓ_1-norm of the parameter vector constitutes a canonical example [Tibshirani, 1996]—are characterized by points of non-differentiability occurring where the parameters are zero, a property that encourages sparse solutions during the optimization process.

From an *optimization* perspective, this non-differentiability introduces an additional layer of difficulty to the already challenging task of optimizing over high-dimensional parameter spaces (note however that these regularization functions are often convex thereby enabling the use of convex optimization techniques). As a result, extensive research at the intersection of machine learning and optimization has addressed the challenge of solving such problems [Beck and Teboulle, 2009, Friedman et al., 2010, Xiao, 2010, Bach et al., 2012, Beck, 2015, Hastie et al., 2015a, Bottou et al., 2018]. Since the primary objective of the learning process is to achieve good generalizing performances on unseen data rather than to compute high-precision solutions to the learning problem, the proposed methods are typically iterative optimization algorithms with low per-iteration computational cost (and slow convergence) allowing approximate solutions to the learning problem to be obtained in reasonable times, even for models involving millions of parameters [Bennett and Parrado-Hernández, 2006, Bottou and Bousquet, 2007, Bottou et al., 2018].

While the learning of preference representations from preference examples is a well-established field in machine learning known as *preference learning* [Fürnkranz and Hüllermeier, 2010b, Wirth et al., 2017, Hüllermeier and Słowiński, 2024b], the potential contribution of using sparsity-inducing regularization functions and associated large-scale optimization methods to address the challenges posed by the learning of utility functions with interactions from decision theory has been little studied so far. Indeed, as in the decision-theoretic litterature, existing approaches in preference learning for learning utility functions such as Choquet integrals, multilinear utilities, or GAI alleviate the computational difficulty by resorting in practice to prior reductions of the parameter space [Tehrani et al., 2012b, Bigot et al., 2012, Tehrani et al., 2014a, Bresson et al., 2021, Atienza et al., 2024] or relaxations of the constraints on the parameters [Kakula et al., 2020a, Tehrani, 2021]. On the other hand, several attempts to control the complexity of the models using sparsity-inducing regularizations have been made [Anderson et al., 2014, Adeyeba et al., 2015, Pinar et al., 2017, de Oliveira et al., 2022]. However, the computational challenge is not truly addressed in these contributions, as the methods are tested on small-scale problems (fewer than 5 viewpoints).

Goal of the thesis. Thus, in this thesis, we aim to develop *scalable* learning algorithms for utility functions with interactions from decision theory, which do not rely on prior reductions of the models flexibility, but instead focus on *sparse* learning of their parameters, allowing the most important interactions to emerge from preference data, ultimately resulting in simple and interpretable models. We also aim to provide these algorithms for various learning scenarios: *passive* (using a preference database), *active* (using selected examples, potentially in interaction with the decision-maker), and *online* (from a stream of examples). In doing so, we also aim to provide the preference learning community with models interpretable and associated with desirable mathematical properties. The presented work typically includes 1) the formulation of a learning problem suited to the utility model and the learning scenario considered, and 2) the development of an optimization method to address the learning problem.

Organization of the manuscript This thesis consists of six chapters, the first of which provides the fundamental concepts of preference modeling and preference learning. The following chapters present the contributions (and the additional background required):

Chapter 2 focuses on the Choquet integral (CI) of marginal utility functions defined for each viewpoint. We show that we can successively learn marginal utilities from properly chosen preference examples, and sparse representations of the capacity parameterizing the CI from raw databases of preference examples. This chapter is based on several publications: [Herin et al., 2022a] for decision-making under uncertainty and [Herin et al., 2022b, 2024c] for multi-criteria/attribute decision-making and the bipolar CI extension.

Chapter 3 considers a large class of preference models that allow viewpoint interactions by mean of a capacity, including CI and the multilinear utility, and introduces a general approach for learning a sparse representation of this capacity. The proposed learning algorithm specifically addresses the computational challenges arising from the combinatorial nature of interactions and is applied to problems involving more than 20 viewpoints. This chapter is based on the following publications: [Herin et al., 2022c, 2023b].

Chapter 4 focuses on the GAI-decomposable utility function model and introduces a learning approach able to identify the factors of interacting viewpoints (as few as possible) and to learn the utility functions defined on these factors. This chapter builds upon and extends the following publication: [Herin et al., 2024b].

Chapter 5 introduces an algorithm for solving a choice problem among a set of alternatives described by multiple criteria by actively learning the parameters of the decision-maker's utility function. This chapter is based on the following publication: [Herin et al., 2024a].

Chapter 6 introduces an online algorithm for learning a sparse representation of the capacity used in the Choquet integral or multilinear utility, designed for decision contexts where preference examples become available sequentially, or involving a large number of preference examples or a large number of criteria. Moreover, we propose a variant making it possible to include normative constraints on the capacity. This chapter is based on the following publication: [Herin et al., 2024d].

Note that the list of publications resulting from the work presented in this thesis is provided in Appendix D.

Chapter 1
Preference Modeling and Preference Learning

Contents

Summary

This chapter first introduces *preference models* within a general framework in which a decision-maker (DM) considers alternatives described by n viewpoints.Then, the preference models of interest here, i.e., *utility functions* accounting for *interactions* between viewpoints, are presented and compared according to their properties. A distinction is made between 1) *totally decomposable functions*, which rely on *aggregation functions* such as the *Choquet integral* or the *multilinear utility* to combine *marginal utility functions* defined on each viewpoint, and 2) *GAI-decomposable functions*, which additively decompose into sub-utility functions attached to groups of interacting viewpoints. Then, we present the standard approaches for *preference elicitation* (methods for calibrating the model's parameters to the DM's value system), before addressing the question from a *machine learning* perspective. To this end, we provide an introduction to *supervised learning* using the framework of *regularized empirical risk minimization* as a general approach to learning models that accurately explain the data while remaining as simple as possible.

Introduction

Preference modeling [Bouyssou and Vincke, 2009] refers to the construction of mathematical or computational models that capture the attitude of individuals or groups in decision-making processes such as choices or rankings of alternatives. More precisely, these preference models reflect the way human beings or entities compare alternatives to determine which one they prefer. These models come in two main forms (formally defined in Section 1.2): either as utility functions that assign numerical scores to alternatives reflecting their attractiveness to the decision-maker, or as preference relations that allow comparing pairs of alternatives. In any case, they enable systematic representation and analysis of preferences, facilitating comparison and evaluation of alternatives, thereby allowing for decision support or automated decision-making. Pioneering works in preference modeling come from various application domains, such as economics where rational customers are traditionally supposed to make choices according to their expected utility [Von Neumann and Morgenstern, 1944] or mathematical psychology [Tversky and Kahneman, 1981] that study the psychological intricacies involved in decision-making cognitive processes. Finally, preference modeling is paired with *preference elicitation*, the objective of which is to align the preference models' parameters with the decision-maker's value system by interacting with her using specifically designed and dynamic questionnaires.

More recently, preference modeling has become ubiquitous in artificial intelligence [Braziunas and Boutilier, 2008, Chevaleyre et al., 2008, Brafman and Domshlak, 2009, Chevaleyre et al., 2010, Domshlak et al., 2011b, Pigozzi et al., 2016, Song et al., 2024], where preference representations are essential for creating autonomous agents capable of making satisfying decisions. These autonomous agents are generally intended to provide personalized services, and often take the form of recommendation systems or virtual assistants. In such contexts, preference representations are typically derived from data such as clicks or likes on social media, or qualitative feedback on proposed recommendations or provided assistance. The task of automatically deriving preference models from data, through machine learning techniques, is referred to as *preference learning* [Hüllermeier and Fürnkranz, 2013]. It involves conceiving algorithms capable of learning from labeled datasets containing preference information such as ratings or rankings of alternatives. By leveraging statistical learning methods and optimization tools, preference learning aims to uncover preference representations, supporting personalized recommendation systems, decision support tools, and automated decision-making processes.

This chapter is organized as follows: In Section 1, we introduce preference models stemming from decision theory, focusing on the models discussed in this thesis. Then, in Section 2, we present standard approaches for preference elicitation. Finally, in Section 3, we present supervised learning and particularly regularized empirical risk minimization as a tool for learning preference models from preference information.

1 Preference Models

In this section, *decision problems* and *preference models* are first formally defined, followed by the presentation of two major families of utility functions allowing for interactions between viewpoints: *totally decomposable* and *GAI-decomposable* utility functions.

1.1 Decision Problem

Despite the wide variety of decision-making situations, a common structure stands out and allows us to formally define a *decision problem* as the combination of the following three components [Perny, 2000]: a set of *alternatives*, a set of n *viewpoints* (or *evaluation dimensions*) w.r.t. which the alternatives are evaluated, and a *problem statement* that may involve *choosing* an alternative, *ranking* the alternatives, or *assigning* the alternatives *to categories*. The semantics associated with the n viewpoints on alternatives can vary depending on the context and may influence how the decision problem is addressed. For this reason, *decision theory* includes several sub-fields associated with different viewpoints' semantics, the main examples of which are outlined below:

Multiattribute decision-making Multiattribute decision-making refers to the general framework where viewpoints are *attributes* describing the alternatives. For instance, if the alternatives are bikes, the attributes could be the brand, the weight, the frame material (steel, aluminum or titanium) and the presence of a basket. In this context, *multiattribute utility theory* (MAUT) [Keeney and Raiffa, 1976, Dyer, 2005] has established conditions on preferences so that they can be modeled by simple multivariate utility functions.

Multicriteria decision-making In multicriteria decision-making [Roy et al., 1985, Roy and Vincke, 1981, Grabisch, 2016b], the n viewpoints (referred to as *criteria*) are defined by n real-valued functions $c_i, i = 1, \ldots, n$ giving the performances (the higher the better) of the alternatives w.r.t. n perspectives. For instance, a bike can be characterized by its performances w.r.t. lightness, aesthetics, robustness, and cost.

Decision-making under uncertainty In the standard setting of Savage for decision-making under uncertainty [Savage, 1954], alternatives (referred to as *acts*) are described by their outcomes (usually payoffs) in the n *states of nature* that consist in an list of states that cover all possible circumstances that might arise and which may have a different impact on the act's outcomes. For instance, alternatives could be national economic policies resulting in different growth rates depending on the future state of the overall economy. In *decision-making under risk*, the probabilities of the states, and therefore the act consequences are represented by probabilistic lotteries [Von Neumann and Morgenstern, 1944].

Multi-agent decision-making Finally, in multi-agent decision-making, alternatives are evaluated or ranked by n agents. For instance, it could be candidates for a scientific prize evaluated by n jury members. In this context, *social choice theory* [Arrow, 1951] studies how collective decisions can be made by aggregating these evaluations or rankings.

Whatever the context, the difficulty behind a decision problem lies in the simultaneous consideration of n distinct viewpoints. For instance, in multi-criteria decision-making, several criteria typically in conflict must be balanced. In decision-making under uncertainty, one has to weigh up various risks, and in collective decision-making divergent opinions have to be taken into account. Decision theory has therefore proposed and studied models to describe how humans or entities solve such problems. These models, called preference models, enable to reveal preferred alternatives, thereby facilitating the formulation of choices and rankings consistent with the DM's value system. While being adjustable to each value system to account for subjectivity, they verify mathematical properties that ensure rationality of decisions. Before formally introducing preference models in the next section, we give useful notations for the sequel.

Notations Let $N = \{1, \ldots, n\}$ denote the set of n viewpoints representing attributes, criteria, states of nature, or agents. Any alternative is represented by a vector $(x_1, \ldots, x_n)$ of consequences where x_i is its consequence w.r.t. to the i^{th} viewpoint. Let X_i denote the set of possible consequences on the i^{th} viewpoint for all $i \in N$ and $\mathcal{X} = X_1 \times \ldots \times X_n$ the set of all possible consequence vectors. Without loss of generality, non-numerical consequences are assumed to be priorly numerically encoded, and therefore, $X_i \subseteq \mathbb{R}, i = 1, \ldots, n$. Also, for any subset of viewpoints $S \subseteq N$, and for any $x \in \mathcal{X}$, x_S refers to the restriction of the consequence vector x to the consequences w.r.t. viewpoints in S, i.e., $x_S \in X_S = \prod_{i \in S} X_i$. In addition, (x_S, x'_{-S}) refers to the compound consequence vector whose consequences w.r.t. viewpoints in S are those of the vector x and the others are those of the vector x'. When S is a singleton (i.e., $S = \{i\}$), the notation (x_i, x'_{-i}) is used.

1.2 Preference Relation and Preference Model

The concept of preference can be formalized mathematically using *binary relations*, defined by:

Definition 1.1 (binary relation). *A binary relation R on a set $\mathcal{X}$ is a subset of the Cartesian product $\mathcal{X} \times \mathcal{X}$. For any $x, x' \in \mathcal{X}$, the assertion $(x, x') \in R$ is denoted by xRx'.*

Various properties are used to describe binary relations. For instance, a binary relation R on $\mathcal{X}$ is:

- *reflexive*, if for any $x \in \mathcal{X}$, xRx
- *transitive*, if for any $x, x', x'' \in \mathcal{X}$,$x''Rx'$ and $x'Rx \Longrightarrow x''Rx$
- *complete*, if for any $x, x' \in \mathcal{X}$, $x'Rx$ or xRx'.

Also, the *asymmetric part* and *symmetric part* of a binary relation R on $\mathcal{X}$, denoted by A and S, can be defined as follows:

- for any $x, x' \in \mathcal{X}$, $xAx' \Longleftrightarrow xRx'$ and not $x'Rx$
- for any $x, x' \in \mathcal{X}$, $xSx' \Longleftrightarrow xRx'$ and $x'Rx$.

A *preference relation*, denoted by $\succsim$, can then be defined as a binary relation reflecting a DM's preferences, i.e., $x \succsim x'$ if and only if the DM considers that " x at least as good as x'". Since for $x' = x$ it is always the case, a preference relation is reflexive. Conversely, any reflexive binary relation could be regarded as the preference relation of a fictitious DM. This yields the following formal definition [Bouyssou and Vincke, 2009]:

Definition 1.2 (preference relation). *A preference relation $\succsim$ on $\mathcal{X}$ is a reflexive binary relation on $\mathcal{X}$. For any $x, x' \in \mathcal{X}$, the assertion $x \succsim x'$ reads "x is at least as good as x'" or "x is preferred to x'".*

The asymmetric part of $\succsim$ is denoted by $\succ$ and $x \succ x'$ reads "x is strictly preferred to x'", while its symmetric part is denoted by $\sim$ and $x \sim x'$ reads "x is indifferent to x'". Then, a *preference model* can be formally defined as follows:

Definition 1.3 (preference model). *A preference model is a function $\Psi : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Any preference model Ψ induces a preference relation on $\mathcal{X}$, denoted by $\succsim_\Psi$, such that for any $x, x' \in \mathcal{X}$:*

$$\Psi(x, x') \geq 0 \Longleftrightarrow x \succsim_\Psi x'$$

where $\succsim_\Psi$ reads "x is preferred to x' according to model Ψ".

A fundamental class of preference models are those based on a *utility function* $U : \mathcal{X} \to \mathbb{R}$ that assigns a overall evaluation to any alternative. Such preference models thus indicate that x is preferred to x' whenever $U(x) \geq U(x')$, i.e., $\Psi(x, x') = U(x) - U(x')$. In this case, solving a choice decision problem boils down to maximizing U over $\mathcal{X}$. In addition, it is important to note that the induced preference relation $\succsim_\Psi$ admits a specific structure as it is a *weak order*, i.e., it is complete and transitive.

Remark 1.1 (weak orders and utility functions). Conversely, if a preference relation $\succsim$ is a weak order and $\mathcal{X}$ is countable (i.e., finite or denumbrable) then there exists a utility function U representing $\succsim$, i.e., such that for any $x, x' \in \mathcal{X}$, $x \succsim x' \iff U(x) \geq U(x')$ [Fishburn et al., 1979, Bouyssou and Vincke, 2009]. Note that when $\mathcal{X}$ is uncountable, additional requirements on $\mathcal{X}$ are needed to show that there exists U representing $\succsim$ [Debreu et al., 1954].

Example 1.1. *Let us consider four friends choosing a restaurant. Each restaurant is described by the four friends' respective desire to dine there, which they expressed by a score on a scale from 0 to 5. In this setting, a basic utility function is the average of the individual scores. The table below gives an example of individual scores and average overall value for two restaurants x and x'.*

	friend 1	friend 2	friend 3	friend 4	$U(x)$
Restaurant x	2.5	1	5	2.5	2.75
Restaurant x'	0	5	1	2	2

Since $U(x) \geq U(x')$, restaurant x is preferred to restaurant x'.

Preference models based on a utility function fall into the general *aggregate and compare* (AC) approach that consists of first assigning values to alternatives through a utility function U and then, for any pair x, x', comparing $U(x)$ and $U(x')$ using any comparison function $g : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ (i.e., $\Psi(x, x') = g(U(x), U(x')))$ [Perny, 2000]. Exchanging the two steps of the AC approach yields a second general class of preference models, the *compare and aggregate* (CA) approach, where pairs of alternatives are first compared according to each viewpoint, and then the n comparisons are aggregated. More formally, in this case, $\Psi(x, x') = h(g_1(x_1, x'_1), \ldots, g_n(x_n, x'_n))$, where $g_i : X_i \times X_i \to \mathbb{R}$, $i = 1, \ldots, n$ are n comparison functions and $h : \mathbb{R}^n \to \mathbb{R}$ is a function aggregating the n preference indices $g_i(x_i, x'_i)$. An example of such a preference model is given below:

Example 1.2. *Let us consider again the four friends of Example 1.1. Suppose now that they did not give scores to the two restaurants x and x' but voted for one of them. Then a basic CA model consists in declaring the preferred restaurant as the one for which a majority of friends voted. This is achieved by taking $g_i(x_i, x'_i) = 1$ if friend i prefers restaurant x to restaurant and x', $g_i(x_i, x'_i) = 0$ otherwise, and using $h(t_1, \dots, t_n) = \sum_{i=1}^{n} t_i - \frac{n}{2}$ $(n = 4)$. Below we give the individual preference indices $g_i(x_i, x'_i)$ for the four friends compatible with their ratings given in Example 1.1, along with the model output:*

	friend 1	friend 2	friend 3	friend 4	$\Psi(x, x')$
Restaurant pair (x, x')	1	1	0	1	1

This model also highlights x as the preferred alternative since $\Psi(x, x') = 1$.

In the case of a CA model, preference relation $\succsim_\Psi$ is usually not transitive and not complete. Furthermore, as shown in Example 1.2, these models not necessarily require the acquisition of quantitative information such as the individual scores used in Example 1.1, and may rely on ordinal information solely. As such, they are considered to require less elicitation effort than preference models based on utility functions, but also to be less rich in the sense that they convey less information.

In the remainder of this thesis, we focus on preference models based on utility functions. Furthermore, to facilitate the interpretation of such representations or their subsequent manipulation (e.g., optimization of the utility function for recommendation purposes), we consider simple decompositions of the utility function. The additive decomposition of the average score used in Example 1.1 serves as a first basic example. However, as it will be illustrated in the following, such a decomposition is limited in its ability to capture complex preferences, and in particular to take into account interactions between viewpoints. The following subsection presents a broad class of models that overcome these limitations, called *totally decomposable* utility functions.

1.3 Totally Decomposable Utility Functions

Totally decomposable utility functions (also known as *decomposable models* [Grabisch et al., 2016, Krantz and Tversky, 1971]) are utility functions that can be decomposed into two types of elements: a set of n *marginal utility functions* defined for each viewpoint on the one hand, and on the other, an *aggregation function* that aggregates them into a overall utility value. An aggregation function can be defined as follows [Grabisch, 2009]:

Definition 1.4 (aggregation function). *A function $F : [a, b]^n \to \mathbb{R}$ is an aggregation function if it is non-decreasing in each argument, i.e., for any $z, t \in [a, b]^n$ such that $z_i \geq t_i$, $i = 1, \ldots, n$ then $F(z) \geq F(t)$.*

Then, totally decomposable utility functions are defined as follows:

Definition 1.5 (totally decomposable utility function). *A totally decomposable utility function $U : \mathcal{X} \to \mathbb{R}$ is a function of the form:*

$$U(x) = F(u_1(x_1), \ldots, u_n(x_n)), \text{ for any } x \in \mathcal{X}, \tag{1.1}$$

where $u_i : \mathbb{R}^n \to [a, b], i = 1, \ldots, n$ and $F : [a, b]^n \to \mathbb{R}$ is an aggregation function.

Functions $u_i : X_i \to \mathbb{R}, i = 1, \ldots, n$ are univariate utility functions describing the attractiveness of consequences $x_i \in X_i, i = 1, \ldots, n$ for the DM, and are thus referred to as *marginal utility functions*. Then, F aggregates these marginal utilities into a global score reflecting the overall attractiveness of the alternatives. By non-decreasingness of F in each argument, this aggregation is such that for any pair of alternatives $x, x' \in \mathcal{X}$, whenever $u_i(x_i) \geq u_i(x'_i), i = 1, \ldots n$ (x is at least as good as x' alternative w.r.t. every point of view), we have $U(x) \geq U(x')$ (x is globally preferred to x'). In other words, if $\succsim_i, i = 1, \ldots, n$ denotes n weak orders defined by $x \succsim_i x' \iff u_i(x_i) \geq u_i(x'_i)$, preference model $\Psi = U(x) - U(x')$ is *monotonic* w.r.t. these orders, i.e.:

Definition 1.6 (monotonic preference model). *For any orders $\succsim_i, i = 1, \ldots, n$, a preference model Ψ is monotonic w.r.t. to these orders if for any $x, x' \in \mathcal{X}$, $x_i \succsim_i x'_i$ for any $i \in N \implies x \succsim_\Psi x'$. Whenever $x_i \succsim_i x'_i$ for any $i \in N$ holds, x' is said to be weakly Pareto-dominated by x.*

1.3.1 Examples of Aggregation Functions

Beyond a common structure, decomposable utility functions can exhibit diverse behavior depending on the aggregation function considered. In particular, each aggregation function is associated with particular mathematical properties and a certain degree of flexibility. Thus, in what follows, we explore major examples of families of aggregation functions, which are models parameterized by a vector w, allowing the encoding of subjective information used in the aggregation, such as the importance given by the DM to the different points of view or to groups of points of view. Note that to avoid overly cumbersome notations, z_i is used to designate $u_i(x_i)$, the marginal utility with respect to viewpoint i.

Weighted sum Denoted by WS_w, the weighted sum simply adds up the marginal utilities weighted by weight vector w as follows:

Definition 1.7 (weighted sum). *For any $z \in [a, b]^n$, $\mathrm{WS}_w(z) = \sum_{i=1}^n w_i z_i$, with $w \in \Delta_n = \{w \in [0,1]^n | \sum_{i=1}^n w_i = 1\}$.*

Although WS can adjust the importance given to each viewpoint through parameter w and thus model diverse preference behaviors, its linearity limits its ability to account for complex preferences. In particular, the weighted sum is not able to model preferences that involve *interactions* (or *synergies*) between viewpoints, as illustrated in the following:

Example 1.3. *The mayor of a large city plans to build a new train station. Four projects A, B, C, D have been proposed, each evaluated according to three viewpoints: (1) connectivity to the metro network, (2) proximity to the city center, (3) economic viability. The evaluations (marginal utilities) w.r.t. each viewpoint expressed on a 0-10 scale are given in Table 1.1 for the four projects.*

	connectivity	*center proximity*	*economic viability*
A	10	0	10
B	10	10	5
C	3	0	10
D	3	10	5

Table 1.1: Evaluations of the four train station projects (Example 1.3).

Given these evaluations, it is highly likely that the mayor will express the following two preferences: $A \succ B$ and $D \succsim C$. Indeed, if the metro station is very well connected, whatever the distance to the city center, a cheap project is better than an expensive one. On the contrary, if the station is not very well connected, the mayor is highly likely to authorize higher spending to bring the station closer to the center so that people can still easily access the train station. However, there exists no weights $w \in \Delta_3$ such that $\mathrm{WS}_w(10, 0, 10) > \mathrm{WS}_w(10, 10, 5)$ and $\mathrm{WS}_w(3, 10, 5) \geq \mathrm{WS}_w(3, 0, 10)$ since it is equivalent to $w_3 > 2w_2$ and $2w_2 \geq w_3$.

Intuitively, there is a kind of redundancy between viewpoints 1 and 2, and thus, when both performances reach their maximum on these viewpoints, the weighted sum ends up rewarding their common quality twice. In contrast, by considering the non-linear model $F(x) = \frac{4}{5}x_1 + \frac{1}{3}x_2 + \frac{1}{5}x_3 - \frac{1}{3}\min x_1, x_2$, this excess utility is corrected through the term $-\frac{1}{3}\min\{x_1, x_2\}$. We then obtain $F(A) = 10 > 9 = F(B)$ and $F(D) = \frac{86}{15} \geq \frac{69}{15} = F(C)$. In fact, F is an instance of the Choquet integral, introduced in the following section.

Remark 1.2 (preferential independence). it can easily be checked that a preference model Ψ based on an additively decomposed utility function verifies *mutual preferential independence* (MPI), i.e., $(x_S, y_{-S}) \succsim_\Psi (x'_S, y_{-S}) \implies (x_S, t_{-S}) \succsim_\Psi (x'_S, t_{-S})$, for any $x, x', y, t \in \mathcal{X}$ and $S \subseteq N$. In words, such a condition means that the preference between two alternatives does not depend on the viewpoints for which they have the same consequences. Thus, another way to understand that WS can not represent preferences $A \succ B$ and $D \succsim C$ in Example 1.3, is to see that they constitute a violation of MPI for $S = \{2, 3\}$. Conversely, if $\succsim$ is a weak order satisfying MPI and some additional technical assumptions it is representable by an additive utility [Bouyssou and Pirlot, 2016].

Remark 1.3 (weak separability). On the other hand, it can easily be checked that, by non-decreasingness of aggregation function F_w, a preference model Ψ based on a totally decomposable utility functions satisfies a weaker independence condition than MPI, referred to as *weak separability* and defined by: $(x_i, y_{-i}) \succ_\Psi (x'_i, y_{-i}) \implies (x_i, t_{-i}) \succsim_\Psi (x'_i, t_{-i})$, for any $x, x', y, t \in \mathcal{X}$ and $i \in N$. Conversely, if $\succsim$ is a weak order and satisfies weak separability, it is representable by a totally decomposable utility function [Bouyssou et al., 2013] (Chapter 16). Note that when $\mathcal{X}$ is uncountable, additional requirements on $\mathcal{X}$ are needed to prove the latter implication.

Example 1.3 illustrates the fact that natural human preferences may involve interactions between viewpoints, which cannot be captured by a linear aggregation function. However, it also suggests that this limitation can be addressed by extending the linear aggregation function with non-linear interaction terms. In the following, we present major examples of general aggregation functions accounting for interactions between viewpoints.

Choquet integral The *Choquet integral* [Choquet, 1954] is a flexible aggregation function initially used in decision-making under uncertainty [Schmeidler, 1989] and more recently used in multicriteria decision-making to model preferences in the presence of interacting criteria [Grabisch, 1996, Grabisch et al., 2009]. Viewpoint interactions are modeled with a weighting system $w : 2^N \to \mathbb{R}$ that attaches a weight $w(S)$ to any possible set of viewpoints $S \subseteq N$, reflecting their relative importance. This weighting system is a *capacity*.

Definition 1.8 (capacity). *A set function $w : 2^N \to \mathbb{R}$ is a capacity if it is monotonic w.r.t. set inclusion i.e., $\forall A \subseteq B, w(A) \leq w(B)$ and it is normalized i.e., $w(\emptyset) = 0$ and $w(N) = 1$.*

Depending on the literature, a capacity can also be called a *non-additive measure* or a *fuzzy measure* [Sugeno, 1974]. The Choquet integral, denoted by C_w, employs capacity w to aggregate marginal utilities as follows:

Definition 1.9 (Choquet integral). *For any $z \in [a, b]^n$,*

$$C_w(z) = \sum_{i=1}^{n} \Big[w(Z_{(i)}) - w(Z_{(i+1)})\Big] z_{(i)} \tag{1.2}$$

$$= \sum_{i=1}^{n} \Big[z_{(i)} - z_{(i-1)}\Big] w(Z_{(i)}) \tag{1.3}$$

where w is a capacity and $(.)$ is any permutation of N such that $z_{(1)} \leq \ldots \leq z_{(n)}$, $z_{(0)} = 0$, $Z_{(i)} = \{(i), \ldots, (n)\}, i = 1, \ldots, n$ and $Z_{(n+1)} = \emptyset$.

In words, $z_{(i)}$ corresponds to the i^{th} highest marginal utility among $(z_1, \ldots, z_n)$ and $Z_{(i)}$ contains the viewpoints for which the marginal utility is greater than or equal to $z_{(i)}$.

Remark 1.4 (marginal utilities commensurability). As the Choquet integral requires ordering marginal utilities $z_i, i = 1, \ldots, n$, they are usually assumed to be *commensurate*, i.e., for any pair of viewpoints $i, j \in N$, $z_i \geq z_j$ means that z_i is at least as good w.r.t. the i^{th} viewpoint as z_j w.r.t. the j^{th} viewpoint.

When capacity w is *additive*, i.e., $w(S) = \sum_{i \in S} w(\{i\})$, it can easily be checked with Equation 1.2 that C_w boils down to a weighted sum with weights $w(\{i\}), i = 1, \ldots, n$. However, w is potentially non-additive, enabling the Choquet integral to account for positive (resp. negative) synergies between viewpoints through *super-additivity*, e.g., for $S = \{i, j\}$, $w(\{i, j\}) > w(\{i\}) + w(\{j\})$, (resp. *sub-additivity*, e.g., $w(\{i, j\}) < w(i) + w(\{j\})$). The following example illustrates the aggregation performed by the Choquet integral on a small example with three viewpoints ($N = \{1, 2, 3\}$):

Example 1.4. *Let us consider an alternative whose marginal utilities (z_1, z_2, z_3) are such that $z_2 \leq z_1 \leq z_3$. Equation 1.3 gives $C_w(z_1, z_2, z_3) = z_2 w(\{1, 2, 3\}) + [z_1 - z_2] w(\{1, 3\}) + [z_3 - z_1] w(\{3\})$. As illustrated by Figure 1.1 (left), this computation first takes the smallest marginal utility (z_2) —achieved for all viewpoints— weighted by the grand coalition weight (red area), and adds the increment $(z_1 - z_2)$ —achieved for viewpoints 1 and 3— weighted by this viewpoint pair's weight (orange area). Finally it adds the last increment $(z_3 - z_1)$ —achieved for viewpoint 3—, weighted by this viewpoint weight (yellow area). The total area can be equivalently computed using Equation 1.2 which, as represented by Figure 1.1 (right), amounts to computing the area with a different subdivision.*

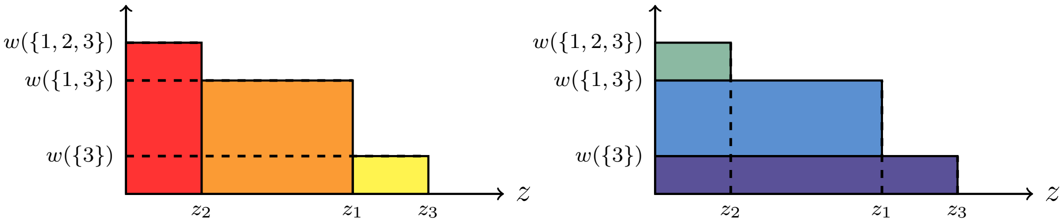


Figure 1.1: Illustration of the Choquet integral computation for three viewpoints.

Remark that when w is additive, i.e., $w(\{1,2,3\}) = w(\{1\}) + w(\{2\}) + w(\{3\}) = 1$ and $w(\{1,3\}) = w(\{1\}) + w(\{3\})$, the area coincides with the integral of the decumulative distribution of a discrete random variable taking values z_1, z_2, z_3 with probabilities $w(\{1\}), w(\{2\}), w(\{3\})$. Such quantity coincides with the expectation of the latter random variable. Indeed we have, $C_w(z_1, z_2, z_3) = z_2(w(\{1\}) + w(\{2\}) + w(\{3\})) + [z_1 - z_2](w(\{1\}) + w(\{3\})) + [z_3 - z_1]w(\{3\}) = w(\{1\})z_1 + w(\{2\})z_2 + w(\{3\})z_3$. Thus, we recover the fact that C_w boils down to the weighted sum when w is additive.

It is important to note that, for any $S \subseteq N$, $C_w((\mathbf{1}_S, \mathbf{0}_{\bar{S}})) = w(S)$, where $\mathbf{0}$ and $\mathbf{1}$ are vectors in $\mathbb{R}^n$ whose components all equal 0 and 1 respectively. Consequently, if the marginal utilities are taken in $[0,1]$, $w(S)$ *coincides with the utility of the alternative which is completely satisfactory (resp. unsatisfactory) according to the viewpoints in S (resp. $\bar{S}$).* In this sense, $w(S)$ can be interpreted as the overall importance of the group of viewpoints S. This overall importance can be further decomposed as a sum of contributions from all possible subgroups, using the *Möbius transform* of w, denoted by m_w, and defined for any $S \subseteq N$ by:

$$m_w(S) = \sum_{T \subseteq S} (-1)^{|S \setminus T|} w(T) \tag{1.4}$$

$$\text{i.e., } w(S) = \sum_{T \subseteq S} m_w(T)$$

For instance, for $S = \{i,j\}, i,j \in N$, $m_w(\{i,j\}) = w(\{i,j\}) - (w(\{i\}) + w(\{j\}))$. Thus, by measuring the gap to additivity, coefficient $m_w(\{i,j\})$ indicates the contribution of the sole interaction between viewpoints i and j to the group importance $w(\{i,j\})$. Remark that $m_w(\{i,j\}) = C_w(\mathbf{1}_{ij}, \mathbf{0}_{\bar{ij}}) - (C_w(\mathbf{1}_i, \mathbf{0}_{\bar{i}}) + C_w(\mathbf{1}_j, \mathbf{0}_{\bar{j}}))$. Thus, in other words, $m_w(\{i,j\})$ quantifies the extra or lost satisfaction due to the joint satisfaction of both viewpoints (compared to the sum of the satisfaction yielded by a separate satisfaction of the two viewpoints).

Depending on their sign, coefficients $m_w(S), S \subseteq N$ (called *Möbius masses*) thus allow revealing positive or negative synergies between viewpoints. Finally, Möbius transform m_w allows for a simple reformulation of the Choquet integral [Chateauneuf and

Jaffray, 1989], as a sum of disjunctive or conjunctive (depending on the sign of the Möbius coefficients) interaction terms:

$$C_w(z) = \textstyle\sum_{S\subseteq N} m_w(S) \min_{i\in S}\{z_i\}, \text{ for any } z \in [a,b]^n. \tag{1.5}$$

Another alternative representation of the capacity is the *interaction indices* representation [Grabisch, 1997b], denoted by I_w and defined as follows:

$$I_w(S) = \sum_{T\subseteq N\setminus S} \frac{(n-|T|-|S|)!|T|!}{(n-|S|+1)!} \sum_{L\subseteq S} (-1)^{|S\setminus L|} w(L\cup T) \text{ for any } S\subseteq N. \tag{1.6}$$

Coefficient $I_w(S)$ can be interpreted as the average contribution of the group of viewpoints S to the importance of the sets containing S. For instance, for $S=\{i,j\}$, $I_w(\{i,j\}) = \sum_{T\subseteq N_{-ij}} \frac{(n-|T|-2)!|T|!}{(n-2+1)!}(w(T\cup\{i,j\}) - w(T\cup i) - w(T\cup j) + w(T))$. When $S=\{i\}$, the interaction index boils down to the *Shapley values* [Shapley, 1971], defined by:

$$\phi_i = \sum_{T\subseteq N_{-i}} \frac{(n-|T|-1)!|T|!}{n!}(w(T\cup\{i\}) - w(T)) \tag{1.7}$$

Equations allowing conversions between the different representations of a capacity [Grabisch et al., 1998] are given in Table 1.2 where B_k denotes the k-th Bernoulli number recursively defined by $B_k = \sum_{l=0}^{k}\binom{k}{l}B_l$, for any $k\in\mathbb{N}$.

	w	m_w	I_w
w	—	$w(S) = \sum_{T\subseteq S} m_w(T)$	$w(S) = \sum_{T\subseteq N}\left[\sum_{k=0}^{\lvert T\cap S\rvert}\binom{\lvert T\cap S\rvert}{k}B_{t-k}\right] I_w(T)$
m_w	Eq. 1.4	—	$m_w(S) = \sum_{T\supseteq S} B_{t-s} I_w(T)$
I_w	Eq. 1.6	$I_w(S) = \sum_{T\supseteq S}\frac{1}{\lvert T\rvert-\lvert S\rvert+1} m_w(T)$	—

Table 1.2: Conversion formulas for the different representations of a capacity.

A case of negative synergy between viewpoints was already illustrated in Example 1.3. The following example showcases real-life positive synergies:

Example 1.5. *Let us consider two conflicting points of view ($N=\{1,2\}$). For instance, it could be rapidity and eco-friendliness regarding transport alternatives, or opinions of two diverging political parties forming a parliamentary majority on potential prime ministers. Let us now consider a set of three alternatives A,B,C whose marginal utilities expressed* *on a 0-20 scale are given in Table 1.3.*

Alternative	viewpoint 1	viewpoint 2
A	0	18
B	19	1
C	10	9.9

Table 1.3: Evaluations of three alternatives w.r.t. two conflicting viewpoints.

As the viewpoints are conflicting, no alternative is fully satisfying. Let us then consider the following preferences: the balanced alternative is strictly preferred to the unbalanced solutions, i.e., $C \succ A$ *and* $C \succ B$*. For instance, travelers with some ecological awareness typically seek a compromise that allows for a reasonable travel time and an acceptable ecological footprint, and political parties forming a coalition have generally agreed beforehand to find a consensus prime minister. Let also assume that the two unbalanced solutions are indifferent to the DM, i.e.,* $A \sim B$*.*

Then, as illustrated by Figure 1.2-left that represents the alternatives according to their marginal utilities, no linear aggregation can associate to C *a strictly higher level of satisfaction while associating to* A *and* B *the same level of satisfaction. However, a positive synergy term* $\alpha \min\{x_1, x_2\}$ *with* $\alpha > 0$*, can be used to grant additional value to alternatives with correct evaluations w.r.t. both viewpoints. In Figure 1.2-right are represented the level curves of an aggregation function fulfilling this condition, i.e.,* $F(z) = \frac{1}{3}(z_1+z_2)+\frac{1}{3}\min\{z_1, z_2\}$*. The balanced solution is assigned an overall value strictly higher than* 8 *while the unbalanced solutions are assigned overall values strictly lower than* 8*.*

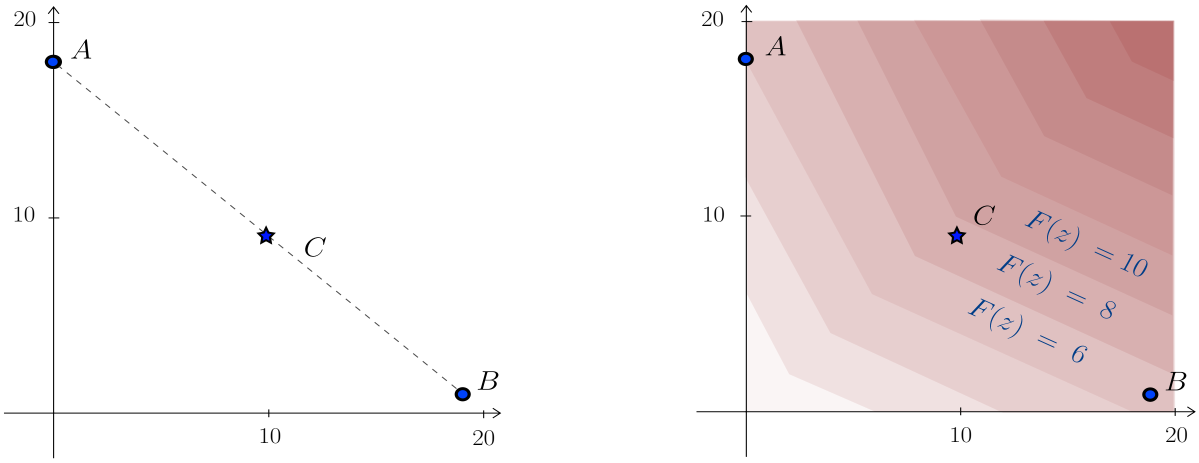


Figure 1.2: A set of alternatives evaluated w.r.t. two conflicting viewpoints (Ex. 1.5).

By Equation 1.5, it can easily be checked that aggregation function F *coincides with a Choquet integral associated with the Möbius transform* $(m_w(\{1\}), m_w(\{2\}), m_w(\{1,2\})) =$

$(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. *This Mobius transform corresponds to the capacity* $(w(\{1\}), w(\{2\}), w(\{1,2\})) = (\frac{1}{3}, \frac{2}{3}, 1)$*, which allows modeling a positive interaction between the two viewpoints through super-additivity, i.e.,* $w(\{1,2\}) - (w(\{1\}) + w(\{2\})) = m_w(\{1,2\}) = \frac{1}{3} > 0$.

Taking into account all possible interactions between viewpoints provides extensive flexibility in preference modeling but requires determining the 2^n coefficients defining the capacity. A frequent option used to control model complexity is to take into account only interactions within groups of limited size, and thus to consider *k-additive* capacities, defined as follows:

Definition 1.10 (*k*-additive capacity). *A capacity w is k-additive if its Möbius transform m_w satisfies $m_w(S) = 0$ for all $S \subseteq N$ such that $|S| > k$, and there exists $S \subseteq N$, such that $|S| = k$ and $m_w(S) > 0$.*

Whenever w is 1-additive, $w(S) = \sum_{i \in S} w(\{i\})$, $S \subseteq N$ and the Choquet integral boils down to a weighted sum. Whenever w is 2 additive, the weighted sum is augmented with a linear combination of pairwise minimum of type $\min\{x_i, x_j\}$ allowing the representation of positive or negative synergies for every pair of criteria:

$$C_w(x) = \sum_{i=1}^{n} m_w(\{i\})x_i + \sum_{i,j} m_w(\{i,j\}) \min\{x_i, x_j\} \tag{1.8}$$

Another particular class of capacities are *symmetric capacities*, defined as follows:

Definition 1.11 (symmetric capacity). *A capacity w is symmetric if for any $A, B \subseteq N$ such that $|A| = |B|$, $w(A) = w(B)$.*

When capacity w is symmetric, we can define a weight w_i such that $w_i = w(S_i) - w(S_{i-1})$, for any sets S_i, S_{i-1} of sizes i and $i-1$. In this case, it can easily be checked using Equation 1.2, that the Choquet integral boils down to an *ordered weighted average* (OWA).

Definition 1.12 (ordered weighted average). *For any $z \in [a,b]^n$ and any permutation (.) of N such that $z_{(i-1)} \leq z_{(i)}$, $i = 1, \ldots, n$, $\text{OWA}_w(z) = \sum_{i=1}^{n} w_i z_{(i)}$, where $w \in \Delta_n$.*

OWA aggregators include simple and well-known aggregation functions such as the min operator $\min_{i \in N}\{z_i\}$ (for $w = (1, 0, \ldots, 0)$), the max operator $\max_{i \in N}\{z_i\}$ (for $w = (0, \ldots, 0, 1)$), and k-order statistics (for $w_k = 1$ and $w_i = 0$ for any $i \neq k$). Note that in the general case, the Choquet integral is an averaging operator, i.e., $\min_{i \in N}\{z_i\} \leq C_v(z) \leq \max_{i \in N}\{z_i\}$ for any $z \in [a,b]^n$ [Marichal, 2000].

Finally, an additional interesting class of capacities is that of *supermodular* capacities:

Definition 1.13 (supermodular capacity). *A capacity w is supermodular (or convex) if $w(S \cup T) + w(S \cap T) \geq w(S) + w(T)$ for any $S, T \subseteq N$.*

The capacity associated to the Choquet integral instance used in Example 1.5 to model the preference for the balanced solution is supermodular since $m_w(\{1,2\}) = \frac{1}{3} > 0$ and thus $w(\{1,2\}) > w(\{1\}) + w(\{2\})$. More generally, supermodular capacities are known to promote solutions with balanced evaluation vectors using the Choquet integral [Chateauneuf and Tallon, 2002b, Lesca and Perny, 2010].

For further properties of the Choquet integral, the interested reader may refer to the several axiomatizations of the Choquet integral, proposed in the context of decision-making under uncertainty [Schmeidler, 1989, Köbberling and Wakker, 2003] or multi-criteria decision-making [Marichal, 2000, Timonin, 2015, Labreuche, 2018]. Let us now present an even more general aggregation function, referred to as the *bipolar Choquet integral*, that allows performing a distinct aggregation of the “good” and “bad” evaluations using distinct capacities.

Bipolar Choquet integral When marginal utility functions are valued in an interval $[a, b]$ endowed with a neutral level $c \in [a, b]$, defining good evaluations (higher than the neutral level) and bad evaluations (lower than the neutral level), the scale is said to be *bipolar*. In such case, the bipolar Choquet integral extends the Choquet integral by using two capacities that cooperate in weighting viewpoints or subset of viewpoints; one applies to the positive part of the evaluation vector whereas the other applies to the negative part [Labreuche and Grabisch, 2006a]. This extension, inspired by Kanheman and Tversky's *cumulative prospect theory* (CPT) [Tversky and Kahneman, 1979] in decision-making under risk, allows the representation of decision behaviors that may vary depending on whether good or bad consequences come into play. Denoted by $BC_{w,w'}$, it is formally defined by:

Definition 1.14 (bipolar Choquet integral). *For any $z \in [a, b]^n$ and two capacities w, w', $BC_{w,w'}(z) = C_w(z^+) + C_w(-z^-)$, where $z^+ = (\max(c, z_i))_{i=1}^n$ and $z^- = (\max(-c, -z_i))_{i=1}^n$.*

it can easily be checked that $C_w(z) = -C_{\bar{w}}(-z)$ for any $z \in [a, b]^n$, where $\bar{w}$ is the dual capacity of w defined by $\bar{w}(A) = 1 - w(N \setminus A)$ for all $A \subseteq N$. Therefore $BC_{w,w'}(z) = C_w(z^+) - C_{\bar{w'}}(z^-)$. This latter formulation makes more explicit the balance between positive and negative arguments where losses are deducted from benefits such as in CPT. Moreover, if $w' = w$, $BC_{w,w'}(z) = C_w(z^+) + C_w(-z^-) = C_w(z)$ and therefore the bipolar Choquet integral boils down to the Choquet integral. The following example illustrates the aggregation performed by BC on a toy example with three viewpoints:

Example 1.6. *Let us consider three viewpoints and an alternative whose marginal utilities w.r.t. the three viewpoints* (z_1, z_2, z_3) *are such that* $z_2 \leq 0 \leq z_1 \leq z_3$ *where* 0 *is a neutral evaluation separating bad from good evaluations. Then we have* $BC_{w,w'}(z) = C_w(z_1, 0, z_3) + C_{w'}(0, z_2, 0)$.

By allowing for a distinct aggregation of good and bad evaluations, the bipolar Choquet integral allows for an even more descriptive power than the Choquet integral. For a more in-depth introduction to the bipolar Choquet integral, interested readers may refer to Labreuche and Grabisch [2006b], Martin and Perny [2021].

Another major example of an aggregation function that allows for interactions, also based on a capacity, is the *multilinear utility*, which we present below:

Multilinear utility The *multilinear utility*, originally proposed in game theory [Owen, 1972], was introduced in decision theory for multiattribute decision-making under risk [Keeney and Raiffa, 1976] and is also axiomatically justified for both multiattribute [Dyer and Sarin, 1979] and multicriteria decision-making [Grabisch et al., 2016, Chap. 6]. Similarly to the Choquet integral, the multilinear utility, denoted by ML_w, allows modeling interactions between viewpoints through the use of a capacity w. The aggregation performed by ML_w is given below:

Definition 1.15 (multilinear utility). *For any* $z \in [a, b]^n$ *and any capacity* w, $\mathrm{ML}_w(z) = \sum_{S \subseteq N} w(S) \prod_{i \in S} z_i \prod_{i \notin S}(1 - z_i)$.

Similarly to the Choquet integral, the multilinear utility can be expressed using the Möbius transform:

$$\mathrm{ML}_w(z) = \textstyle\sum_{S \subseteq N} m_w(S) \prod_{i \in S} z_i \tag{1.9}$$

Equation 1.5 and Equation 1.9 reveal that the multilinear utility has a form similar to the Choquet integral, with the interaction terms $\min_{i \in S}\{z_i\}$ being substituted by product interaction terms $\prod_{i \in S} z_i$. In the following example, we illustrate how this difference may impact preference modeling.

Example 1.7. *Let us consider a toy case involving two conflicting viewpoints and a set of non Pareto-dominated alternatives, yielding the Pareto front represented in Figure 1.3.*

Among them, we have highlighted the alternatives for which there exists a weight w that makes it the optimal alternative, with the Choquet integral C_w (on the left in yellow or black), with the multilinear utility ML_w (on the right graph in red or black), and with the weighted sum WS_w (in black only on the two graphs). We observe that the Choquet integral and the multilinear utility enable accessing different sets of alternatives. Also, the higher descriptive power of models with interactions compared to the weighted sum remains clear here, as the weighted sum is again limited to modeling preference for the unbalanced alternatives constituting the convex hull of the alternative set (black points).

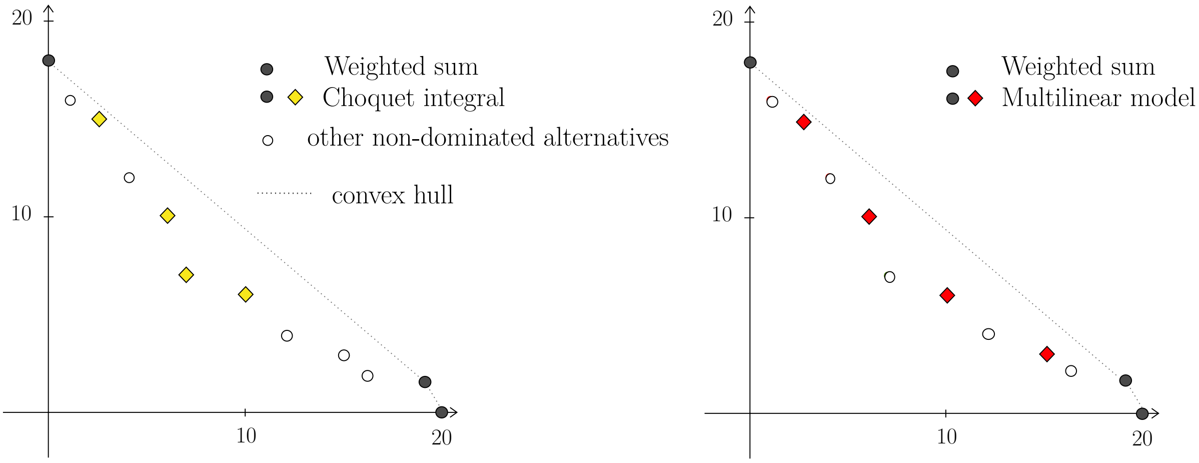


Figure 1.3: Alternatives accessible by maximizing WS_w, C_w and ML_w.

Note that both the Choquet integral C_w and multilinear utility ML_w, when defined on the unit hypercube $[0,1]^n$, are extensions of the capacity w on this same hypercube. Indeed, the vertices of the unit cube correspond to the vectors $(\mathbf{1}_S, \mathbf{0}_{-S}), S \subseteq N$, and it can easily be checked that $C_w((\mathbf{1}_S, \mathbf{0}_{-S})) = \mathrm{ML}_w((\mathbf{1}_S, \mathbf{0}_{-S})) = w(S)$. More precisely, the Choquet integral corresponds to a parsimonious (i.e., using the fewest possible number of vertices) linear interpolation, while the multilinear utility ML_w corresponds to an interpolation that uses all possible vertices [Singer, 1985, Grabisch et al., 2016]. The two distinct interpolations defining C_w and ML_w are represented in Figure 1.4(left) and Figure 1.4(right) for $n = 2$ and a capacity w defined by $w(\{1\}) = w(\{2\}) = 0.2$.

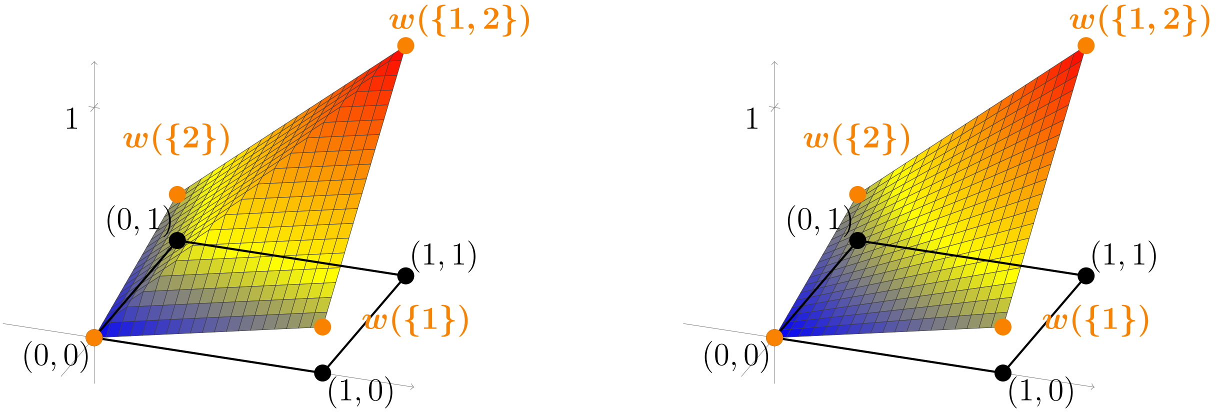


Figure 1.4: C_w (left) and ML_w (right) as two distinct interpolations of w on the unit square.

As illustrated in Example 1.7, despite the enhanced descriptive power of the Choquet integral or the multilinear utility compared to the weighted sum, some non Pareto-dominated alternatives may not be accessible by maximizing $C_w(z)$ or $\mathrm{ML}_w(z)$. In contexts where no prior information about the preference system of the DM is available, it may be the case that any non Pareto-dominated alternative is of possible interest and must be accessible by the aggregation function. In this case, the standard approach is to use a *weighted Chebyshev norm.*

Weighted Chebyshev norm The *weighted Chebyshev norm* (also known as the weighted infinite norm) defined by $\|z\|_{w,\infty} = \max_{i\in N}\{w_i|z_i|\}$ for any $z \in [a,b]^n$ and $w \in \Delta_n$, makes it possible to measure distances between solutions by taking into account the importance attributed to the viewpoints through weights $w_i, i = 1,\dots,n$. Thus, it can be used to quantify the overall quality of a solution in a set X_P of non Pareto-dominated solutions as its proximity to the *ideal point* (the fictitious alternative with maximal marginal utilities w.r.t. every viewpoint) [Wierzbicki, 1986]. This yields the following scalarizing function, denoted by T_w, and defined by:

Definition 1.16 (weighted Chebyshev distance). *For any $z \in X_P \subseteq [a,b]^n$, $T_w(z) = -\max_{i\in N}\{w_i|z_i - I_i|\}$, where $w \in \Delta_n$ and I is the ideal point whose coordinates I_i are defined by $I_i = \max_{z\in X_p}\{z_i\}$ for any $i \in N$.*

By adjusting weights $w_i, i = 1,\dots,n$, a wide range of preference behaviors can be modeled. More formally, for any non Pareto-dominated alternative $z \in X_P$, there exists a weight vector $w \in \Delta_n$ such that z is the preferred alternative, i.e., $z = \arg\max_{z'\in X_P} T_w(z')$ [Wierzbicki, 1986]. Therefore, contrarily to the Choquet integral or multilinear utility, the weighted Chebyshev distance allows access to every single point of the Pareto front of Figure 1.3. However, as it is illustrated in the following example, this does not imply that $T_w(z)$ can capture any preference order within X_P.

Example 1.8. *Let us consider a set X_P of four non Pareto-dominated alternatives A, B, C, D whose marginal utilities w.r.t. three viewpoints are given below.*

	viewpoint 1	viewpoint 2	viewpoint 3
A	1	0.5	1
B	0.5	1	1
C	1	0.5	0
D	0.5	1	0

As preferences $A \succ B$ and $D \succ C$ violate mutual preferential independence for $S = \{1, 2\}$, the order $A \succ B \succ D \succ C$ can not be represented using a weighted sum (see Remark 1.2). Furthermore, the ideal point is $I = (1, 1, 1)$ and thus preferences $A \succ B$ and $D \succ C$ are representable with T_w if and only if $T_w(1, 0.5, 1) > T_w(0.5, 1, 1)$ and $T_w(0.5, 1, 0) > T_w(0.5, 1, 0)$, which is in turn equivalent to $w_1 > w_2$ and $\max\{\frac{1}{2}w_2, w3\} > \max\{\frac{1}{2}w_1, w3\}$ and is thus contradictory. Then such order cannot be described with a weighted Chebyshev norm. However, the Choquet integral $C_w(z) = \frac{4}{10}z_2 + \frac{6}{10}\min\{z_1, z_2\}$ allows representing both preferences $A \succ B$ and $D \succ C$. Indeed $C_w(1, 0.5, 1) = \frac{13}{10} > \frac{12}{10} = C_w(0.5, 1, 1)$ and $C_w(0.5, 0, 0) = \frac{4}{10} > \frac{2}{10} = C_w(1, 0.5, 0)$.

Finally, we briefly present the *Sugeno integral*, often regarded as the ordinal counterpart of the Choquet integral:

Sugeno integral Similarly to the Choquet integral, the *Sugeno integral* [Sugeno, 1974] was initially used in decision-making under uncertainty [Dubois et al., 1998] and exploited later in multicriteria decision-making [Grabisch and Labreuche, 2010]. Denoted by S_w, it also employs a capacity w to aggregate the marginal utilities so as to account for viewpoint interaction. However, it differs from the Choquet integral in that the arithmetic operations $(+, \times)$ are replaced by $(\max, \min)$ operations:

Definition 1.17 (Sugeno integral). *For any $z \in [0, 1]^n$ and any permutation (.) of N such that $z_{(i-1)} \leq z_{(i)}$, $i = 1, \ldots, n$, $Z_{(i)} = \{(i), \ldots, (n)\}$, $S_w(z) = \max_{i \in N} \min\{z_{(i)}, w(Z_{(i)})\}$, where w is a capacity. Note that marginal utilities and capacity values have to be expressed on the same scale.*

In the next section, we present a highly flexible model that generalizes totally decomposable utility functions, referred to as *GAI-decomposable utility functions*.

1.4 GAI-decomposable Utility Functions

GAI-decomposable utility functions, where GAI stands for *Generalized Additive Independence*, have been proposed in multiattribute utility theory by Fishburn [1970] as a generalization of additive utility functions. In words, it is an additive decomposition of the utility function in multivariate terms that reflect interactions between viewpoints. It can be formally defined as follows:

Definition 1.18 (GAI-decomposable utility). *Let $\mathcal{F}$ be a collection of possibly overlapping subsets of N. A utility function $U : \mathcal{X} \to \mathbb{R}$ is GAI-decomposable w.r.t. to $\mathcal{F}$ if there exists functions $u_S : X_S \to \mathbb{R}, S \in \mathcal{F}$ such that :*

$$U(x) = \sum_{S \in \mathcal{F}} u_S(x_S), \text{ for any } x \in \mathcal{X}$$

Obviously, any real-valued function $U : \mathcal{X} \to \mathbb{R}$ can be regarded as a GAI-decomposable function by taking $\mathcal{F} = \{N\}$ and $u_N(x) = U(x)$. Also, by Equations 1.5 and 1.9, both the multilinear utility and the Choquet integral of marginal utilities can be recovered by taking $\mathcal{F} = 2^N$, and for any $S \subseteq N$, $u_S(x_S) = m_w(S) \prod_{i \in S} u_i(x_i)$ and $u_S(x_S) = m_w(S) \min_{i \in S}\{u_i(x_i)\}$ respectively.

However, by making no assumption on the kind of interaction and allowing completely general interaction terms u_S, GAI-decomposable utility function can be much more flexible than totally decomposable functions. In particular, they allow for the representation of complex preferences involving interactions between viewpoints, even in cases where weak separability does not hold (see Remark 1.3) and therefore totally decomposable models fail. This is illustrated by the following example:

Example 1.9. *Let us take the example of a DM selecting a holiday rental house based on the following attributes: (1) localization type: {seaside, city, high mountains}, (2) room number: {1,2,3}, (3) fun feature: {pool, jacuzzi, sauna}, and (4) neighbor proximity: {isolated, moderate distance, close}. She is likely to prefer a swimming pool to a sauna for a house by the sea, whereas she will prefer a sauna in high mountains. Similarly, she is likely to choose not to be stuck next to her neighbor in high mountains, while preferring to stay in a dense area such as an historic center, and avoid empty industrial zones, when staying in the city. Then, the four following preferences are likely to be encountered:*

- *$P1$: (seaside, 2, pool, moderate distance) $\succ$ (seaside, 2, sauna, moderate distance);*

- *$P2$: (high mountains, 2, sauna, moderate distance) $\succ$ (high mountains, 2, pool, moderate distance);*

3 1	pool	jacuzzi	sauna
seaside	1	0.8	0.5
city	1	0.5	0.2
high mountains	0.5	1	1

Table 1.4: Values of $u_{1,3}$.

4 1	isolated	moderate distance	close
seaside	1	0.8	0.5
city	0	0.5	1
high mountains	1	0.5	0

Table 1.5: Values of $u_{1,4}$.

- *P3: (high mountains, 2, jacuzzi, isolated)* $\succ$ *(high mountains, 2, jacuzzi, close);*

- *P4: (city, 2, jacuzzi, close)* $\succ$ *(city, 2, jacuzzi, isolated)*

However, no totally decomposable utility function can account for these preferences, since P1 and P2 on the one hand, and P3 and P4 on the other, constitute violations of weak separability (see Remark 1.2) for $i = 3$ *and* $i = 4$ *respectively. In other words, preferences over the elements {pool, jacuzzi, sauna} on the one hand, and {isolated, moderately distant, close} on the other, both depend on the location of the house. However, preferences* $P1$*,* $P2$*,* $P3$ *and* $P4$*, can be captured by a GAI-decomposable utility function accounting for the interaction between the localization and the fun feature on one hand, and between the localization and the neighbor proximity on the other. For instance, it can easily be checked that the GAI function* $U(x) = u_{1,3}(x_1, x_3) + u_{1,4}(x_1, x_4)$ *with* $u_{1,3}$ *and* $u_{1,4}$ *defined in Table 1.4 and Table 1.5 is compatible.*

Finally, we conclude this section by presenting in Figure 1.5 a way to identify the family of utility functions one may want to consider, depending on whether the preferences (those we wish to describe with the model, or those we aim to construct using the model) satisfy mutual independence and weak separability.

2 Preference Elicitation

Preference elicitation [Tversky, 1977, Boutilier et al., 1997, Mousseau and Pirlot, 2015] involves interacting with the DM to collect information allowing an accurate modeling of her preferences. In particular, the objective is to align with the DM's value

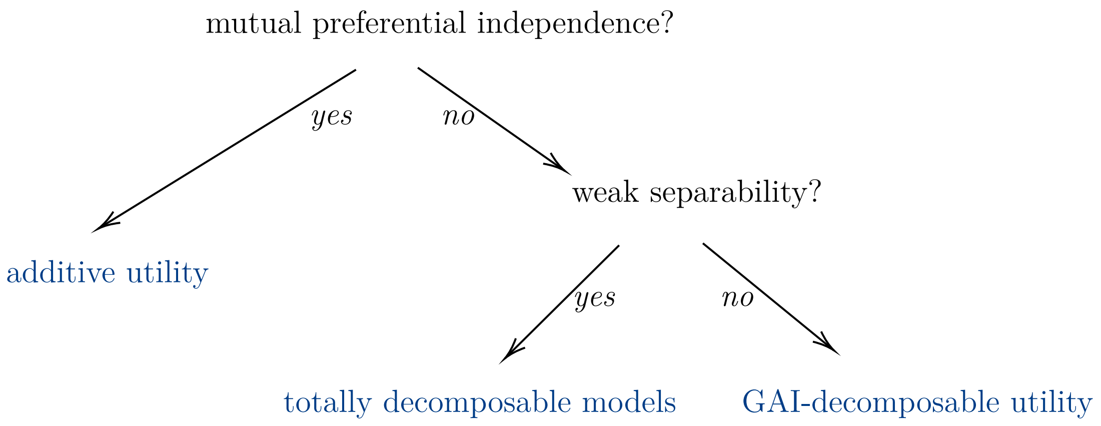


Figure 1.5: Utility function models depending on verified conditions on preferences.

system the parameters of the preference models introduced in the previous section, i.e., marginal utilities $u_i, i = 1, \ldots, n$ and parameter w of the aggregation functions for totally decomposable utility function, or GAI factors $u_S, S \subseteq N$ for GAI-decomposable utility functions. This is generally achieved using *questionnaires* that ask for the ranking or rating of alternatives, followed by the search for parameters consistent with the responses using some *operations research techniques* such as *linear programming*. In this section, we give a brief overview of standard preference elicitation approaches for both totally decomposable models and GAI-decomposable utility functions.

2.1 Marginal Utilities Elicitation

In the framework of the additive utility, marginal utility functions u_i are traditionally elicited by incrementally constructing *standard sequences* of points $(x_i^k, u_i(x_i^k))_{k=1}^q$ on the utility curve [Krantz and Tversky, 1971, Von Winterfeldt and Edwards, 1986, Bouyssou, 2000]. As an illustration, let us consider two viewpoints, and two reference points $x^0 = (x_1^0, x_2^0)$ and $x^1 = (x_1^1, x_2^1)$ where u_1, u_2 are arbitrarily fixed, i.e., $u_1(x_1^0) = u_2(x_2^0) = 0$ and $u_1(x_1^1) = u_2(x_2^1) = 1$. Then, a standard sequence for u_1 can be obtained iteratively, by asking at each iteration $k \geq 2$, "what is the consequence x_1^k such that $(x_1^k, x_2^0) \sim (x_1^{k-1}, x_2^1)$?". Hence, we have that $u_1(x_1^k) = 1 + u_1(x_1^{k-1})$, i.e., $u_1(x^k) = k$, for any $k \geq 1$. However, since each query relies on the answer of the previous query, standard sequences methods suffer from noise propagation [Blavatskyy, 2006]. In multicriteria decision-making, a more robust approach is the UTA (UTility Additive) [Jacquet-Lagreze and Siskos, 1982] that elicits marginal utilities in the additive value model by performing an ordinal regression from a set of priorly acquired pairwise preference or indifference statements.

When DM's preferences are modeled using a Choquet integral or a multilinear aggregation with a capacity w, the indifference statements used in the standard sequences of the form $(x_1^k, x_2^0) \sim (x_1^{k-1}, x_2^1)$ translate into an equation that now involves $w(\{1\})$, $w(\{2\})$, $u_1(x_1^k)$, and $u_1(x_1^{k-1})$. Thus, since $w(\{1\})$ and $w(\{2\})$ are unknown, these equations do not allow us to deduce $u_1(x_1^k)$ from $u_1(x_1^{k-1})$, contrary to the case of the additive utility. A standard approach to circumvent this latter issue is to proceed in two steps: the marginal utilities are first elicited using specific queries that disentangle the respective impact of the marginal utility functions and the capacity, and then the capacity is determined.

A standard method for marginal utilities elicitation in multicriteria decision-making, is the *Macbeth* method [Bana e Costa and Vansnick, 1997]. This method relies on direct queries of utility values $u_i(x_i)$ for some consequences x_i, and pairwise preference intensity queries (e.g., "is the difference of attractiveness between alternative a and b weak or strong?"). In decision-making under risk, some elicitation protocols under the form of standard sequences have been proposed [Wakker and Deneffe, 1996, Abdellaoui, 2000] for the RDU model (Rank Dependent Utility) [Quiggin, 2012] and the CPT model [Kahneman and Tversky, 1979], which are respectively specific instances of the Choquet integral and bipolar Choquet integral for capacities defined as monotone transforms of probability measures. Marginal utilities within the multilinear model can also be elicited using standard sequences in decision-making under uncertainty [Keeney et al., 1993]. They can alternatively be derived from comparisons of preference intensities (e.g., "Is the difference of attractiveness between alternative a and b higher than between alternative c and d?") in multicriteria/multiattribute decision-making [Grabisch, 2016b].

Thus, existing approaches for eliciting marginal utilities in the totally decomposable model with non-linear aggregation rely either on queries that the DM may struggle to answer (e.g., queries on utility values in MACBETH) or on standard sequences that are sensitive to response errors. Furthermore, approaches relying on standard sequences are only formulated for the multilinear model or specific instances of the Choquet integral (e.g., RDU, CPT). This motivates us to introduce a regression-based method for eliciting in a noise tolerant manner the marginal utilities within the general Choquet integral model (or even more generally, within the bipolar Choquet integral model) in *Chapter 2* of this thesis.

Remark 1.5 (simultaneous elicitation of utilities and capacities). Other contributions propose to determine simultaneously the marginal utility functions and the capacity. However, simultaneous optimization of both parameters yields non-convex optimization problems with quadratic constraints involving products of variables $u_i(x_i) \times w(S)$. Some heuristics to solve this problem were proposed. A stochastic method was introduced by [Angilella et al., 2004, 2015], and [Goujon and Labreuche, 2013, Goujon, 2018] discussed a fixed-point method where the problem is split into two iterative linear tasks.

Another heuristic based on a linear approximation of the product of the marginal utility functions with interaction indices was considered by [Galand and Mayag, 2017a]. Additional approaches that leverage machine learning tools are discussed in Section 3.2.2.

2.2 Aggregation Function Parameter Elicitation

After the elicitation of the marginal utilities, the determination of the parameter w of aggregation function F_w has to be addressed. Two types of approaches can be distinguished: elicitation with the aim of solving a specific decision problem and elicitation with the aim of determining a model that is a good representation of the DM's preferences in general.

2.2.1 Decision-focused Elicitation

Decision-focused elicitation methods aim at collecting information about parameter w with the objective of solving a specific decision problem (usually a choice between a set of alternatives). This setting is therefore highly related to *preference-driven (or interactive) multi-objective optimization* [Wierzbicki, 2005] where the goal is to reveal alternatives of interest for the DM within a set of non Pareto-dominated alternatives. Two main families of approaches can be further distinguished:

The local and interactive judgment approach: An initial vector of parameters w is chosen, an optimal solution for F_w is calculated, and then w is allowed to evolve according to user feedbacks until a solution satisfying the DM is reached. This approach, widely used in interactive multicriteria optimization [Steuer, 1986, Vanderpooten and Vincke, 1989], allows a user-driven exploration of the Pareto set, alternating phases of calculation of the current optimal solution and phases of dialogue with the user during which w is updated. It may require numerous interactions, and the quality of the solution chosen at the end of the process is only validated by the DM's instant sense of satisfaction.

Incremental preference elicitation: *Incremental preference elicitation* refers to a set of methods that incrementally increase the knowledge on parameter w and stop as soon as this knowledge is sufficient to solve the decision problem (for example, when a necessarily preferred alternative emerges). A first approach to incremental elicitation consists in progressively reducing the space of admissible values for parameter w. Iteratively, a preference query is chosen, the answer to which induces a new constraint on the space of admissible values for w. Thus, the set of parameter values compatible with the constraints induced by the expressed preference judgments is progressively reduced until the point where an alternative proves optimal (or near optimal) for all remaining admissible parameter values. This approach is introduced in the ISMAUT method [White et al., 1984]. A principle of active question selection is often used, based on the minimization of maximum regret, to choose the most informative question [Wang and Boutilier, 2003, Boutilier et al., 2006, Benabbou et al., 2017a] and derive a robust recommendation. Another approach, more tolerant to noisy responses, is to manage a probability distribution over the parameter space and revise it according to the answers to questions, to choose a decision having the maximum expected value [Chajewska et al., 2000] or minimizing the expectation of regret [Bourdache et al., 2019a]. This type of approach can also be adapted to other uncertainty models such as possibility theory [Adam and Destercke, 2024].

These methods are question-saving, as they direct the questionnaire towards the resolution of a particular instance of decision problem. In the same spirit than incremental preference elicitation, *robust ordinal regressions* approaches [Greco et al., 2008, Angilella et al., 2010, Corrente et al., 2013, Gilbert et al., 2025] aim at using the whole set of parameter values compatible with the observed preferences to formulate robust recommendations, i.e., consistent with all admissible parameters values. Thus, these two types of methods do not determine a specific preference parameter w. Therefore, they are generally not sufficient to obtain representations of DM's preferences or to solve a decision problem involving a new set of alternatives. In the following section, we present another type of approach, which is the one adopted in this thesis, aimed precisely at *identifying from a preference database, a parameter w that is a good representation of the DM's preferences.*

2.2.2 Regression-based Elicitation from a Database of Preference Statements

An important stream of work developed in the literature on multicriteria decision-making concerns the use of regressions for the identification of the capacity parameterizing the Choquet integral, assuming the marginal utilities are known [Grabisch et al., 2008, Grabisch and Labreuche, 2010, Beliakov and Wu, 2019b, 2021]. These regressions are formulated as optimization problems on the set of admissible parameter values, where either the deviation from the observed utility values or the violation of the constraints induced by preferential information such as pairwise preference examples, is minimized (in the latter case they are called *ordinal regressions*).

In particular, we can mention least squares regression with examples of alternatives utility values prescribed by the DM where the objective is to minimize the average squared error with the given values [Murofushi and Sugeno, 1989]. Alternatively, *maximum margin* problems have been formulated to search for parameters that maximize the utility gap between two alternatives involved in a pairwise preference example using linear programming [Marichal, 2000]. Some recent contributions using regression also exist for the multilinear model [Pelegrina et al., 2018, 2020a].

Capacities being defined by $2^n - 1$ parameters in general, the aforementioned optimization problems quickly become intractable as n increases beyond a dozen, whereas the available preference data may not require such a complex model to be well described. Additionally, such a high number of parameters severely limits the interpretability of the model. A common way to overcome these issues is to use k-additive capacities (see Definition 1.10) [Grabisch et al., 2008, Galand and Mayag, 2017b, Ah-Pine et al., 2018, Pelegrina et al., 2020a] ($k = 2$ being the most common choice). Similar restrictions exist for limiting interactions with the notion of *k-interactivity* [Beliakov and Wu, 2019b] or *k-maxitivity* [Beliakov and Wu, 2021]. However, such prior restrictions require arbitrarily setting the maximum size of allowed interactions k and significantly limit the model flexibility.

Consequently, there is a need for advanced regression techniques to identify capacity parameters that remain tractable as n increases and that yield simple and interpretable models, without relying on prior restrictions of model flexibility based on cardinality, such as k-additivity. To develop such approaches, we propose to explore the problem of determining the capacity from a database of preference statements from the perspective of *machine learning* and *optimization* whose intersection provides a framework to formulate parameter learning problems based on statistical theory, while also developing optimization methods for efficiently solving these problems.

2.3 GAI Elicitation

The construction of a GAI-decomposable utility function from preference information requires the determination of the relevant factors to be used in the decomposition (i.e., collection $\mathcal{F}$ of subsets of N) as well as the determination of sub-value functions on these factors (i.e., $u_S(x_S)$ for any $S \in \mathcal{F}$). Some contributions focus on the elicitation of these sub-utility functions, assuming the decomposition of the utility function into factors is known [Gonzales and Perny, 2004, 2005, Braziunas and Boutilier, 2005, Braziunas, 2012]. The proposed methods involve constructing subutilities factors $u_S(x_S)$ in a specific order to exploit conditional independence and then rescaling them to align with previously determined subutilities factors. The construction of each subfactor rely on pairwise indifference queries (e.g., "what can be changed in the consequences of alternative a to make you indifferent between alternatives a and b?"). Some other contributions tackle the problem of learning the decomposition. For instance, a procedure to determine a monotonic well-formed GAI-decomposition from *pairwise preference queries* (e.g., " Is alternative a at least as good as alternative b?") was recently proposed [Grabisch et al., 2022]. In this work, interactions are limited to pairs of viewpoints.

Therefore, all the above-mentioned contributions either assume that the structure of the GAI decomposition is known or that it is limited to interactions involving very few attributes. The construction of a GAI-decomposable utility function with no prior assumption on the size of the interacting groups of viewpoints from preference information thus remains to be addressed. Once again, this thesis aim at leveraging tools from the field of machine learning and optimization to address this challenge. For this reason, we present in the next section the general framework of supervised learning and the optimization methods commonly used to address the formulated learning problems, as well as the specificities of learning preference models, a question that sparked the development of a whole domain in machine learning known as *preference learning*.

3 Preference Learning

In *preference learning* or *preference-based machine learning* [Fürnkranz and Hüllermeier, 2010b, Domshlak et al., 2011a, Busa-Fekete and Hüllermeier, 2014, Wirth et al., 2017, Hüllermeier and Słowiński, 2024a,b], the goal is to *learn* preference models from preference information. This field is part of a broader discipline, *machine learning*, which aims to design algorithms that can *learn from data* and *make predictions*. Machine learning algorithms usually rely on mathematical or computational models whose parameters are optimized to achieve accurate prediction on data. When data is labeled (i.e., it consists of input data tagged with the true output), learning algorithms are called *supervised learning* algorithms. Otherwise, when the algorithm itself has to identify useful labels for prediction, they are referred to as *unsupervised learning* algorithms. As preferential information can be regarded as labeled data, preference learning algorithms typically fall into the first of these categories. For instance, examples of alternative utility values or pairwise comparisons consist of alternatives or pairs of alternatives (inputs) associated with ratings or rankings (labels). Therefore, this section is naturally organized in two parts: the general framework of supervised learning is first presented, and then, preference learning is specifically discussed with a focus on preference models derived from decision theory presented in Section 1.

More precisely, supervised learning is first presented as the formulation of a *regularized empirical risk minimization* (RERM) problem, and then an important focus is put on the optimization method to solve the RERM problem. In this regard, notions of (*convex*) *optimization* are included throughout the section. The emphasis on the optimization aspect is motivated by the fact that, as underlined at the end of the previous section, a main concern is to derive *computationally efficient* methods to learn preference models (e.g., capacities defined by an exponential number of parameters). Concerns relative to the *learning theory* such as "what is the worst prediction error I could do on a new example if I learn a model with t examples?" or conversely "how many inputs examples are needed to correctly learn a model?" are not discussed in this introduction. The reader interested in these questions may refer for instance to the introduction [Bousquet et al., 2003] or the books [Vapnik, 1995, Devroye et al., 1996, Shalev-Shwartz and Ben-David, 2014].

Notations The reader is assumed to be familiar with the basic notions of probability theory, as well as with the Bachman-Landau notations O, θ, Ω. Also, by convention, $\mathbb{1}_{\{C\}}$ equals 0 when condition C is met and 0 otherwise.

3.1 Supervised Learning Framework

3.1.1 Problem Formulation

The starting point of a supervised learning algorithm is a labeled dataset $D = \left\{\left(x^\ell, y^\ell\right)\right\}_{\ell=1}^t$ that contains t samples of *inputs*, denoted by x^ℓ, and their *labels*, denoted by y^ℓ. The inputs are multi-dimensional vectors $x = (x_1, \ldots, x_n)$ belonging to an input space denoted by $\mathcal{X}$, typically taken as $\mathbb{R}^n$. Their corresponding labels are elements of an output space denoted by $\mathcal{Y}$, the definition of which varies depending on the context. When $\mathcal{Y} = \mathbb{R}$, the learning task is referred to as a *regression* task, while when $\mathcal{Y}$ is a finite set of the form $\mathcal{Y} = \{y_1, \ldots y_K\}$, the learning task is referred to as a *classification* task and elements of $\mathcal{Y}$ denotes *class* membership. In the classification setting, the *binary classification* that uses two classes only (e.g., $\mathcal{Y} = \{-1, 1\}$) can be distinguished from the *multiclass classification* that uses $K > 2$ classes. Finally, the training examples in D are regarded as realizations of t random variables $(X^\ell, Y^\ell), \ell = 1, \ldots, t$, assumed to be independent and identically distributed (i.i.d.) according to a joint distribution $\mathcal{D}$ defined over $\mathcal{X} \times \mathcal{Y}$, describing the chances of encountering particular pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ in real life.

Then, the goal of a supervised learning algorithm is to exploit dataset D to find the model h in a *hypothesis class* $\mathcal{H}$ that allows to best predict the label value y of an instance x. More precisely, a hypothesis class is a set of functions $h : \mathcal{X} \to \mathcal{Y}$ that assign label $h(x)$ to any input $x \in \mathcal{X}$. The quality of a prediction is then assessed through a *loss function* $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ that measures the discrepancy between a prediction $h(x)$ and a true label y. Loss functions can take various forms depending on the learning task. For instance, for regression tasks standard choices are the *squared loss* $l(h(x), y) = \frac{1}{2}(h(x) - y)^2$ or the *absolute loss* $l(h(x), y) = |h(x) - y|$. For binary classification tasks, standard choices are the *0-1 loss* $l(h(x), y) = \mathbb{1}_{\{h(x)\neq y\}}$ and the *hinge loss* $l(h(x), y) = \max\{0, 1 - h(x)y\}$ or the *logistic loss* $\log(1+e^{-yh(x)})$ for classifiers of the form $\text{sign}(h(x))$ based on a regression function $h : \mathbb{R}^n \to \mathbb{R}$. Then, when loss function l is specified, the model $h \in \mathcal{H}$ that has the smallest chance of making a wrong prediction is the one that minimizes the *true risk*, denoted by $R(h)$ and defined as the loss expectation w.r.t. $\mathcal{D}$, i.e.,:

Definition 1.19 (true risk). *For any $h \in \mathcal{H}$,*

$$R(h) = \mathbb{E}_{(X,Y)\sim\mathcal{D}}[l(h(X), Y)] = \int_{\mathcal{X}\times\mathcal{Y}} l(h(x), y)p(x, y)dxdy$$

where p is the probability density function of distribution $\mathcal{D}$.

In words, the true risk represents the average prediction error made by a model on all possible data. However, distribution $\mathcal{D}$ is typically unknown, making the true risk impossible to compute. A way to bypass this issue is to compute the *empirical risk* on the available data D, denoted by $R_{emp}(h, D)$, and defined as the average loss on D:

Definition 1.20 (empirical risk). *For any $h \in \mathcal{H}$ and dataset $D = \left\{\left(x^\ell, y^\ell\right)\right\}_{\ell=1}^t$, $R_{\text{emp}}(h, D) = \frac{1}{t}\sum_{\ell=1}^t l(h(x^\ell), y^\ell)$. Let also define, for any $h \in \mathcal{H}$, the random variable $R_{\text{emp}}(h) = \frac{1}{t}\sum_{\ell=1}^t l(h(X^\ell), Y^\ell)$.*

The task of minimizing the empirical risk over a hypothesis class is referred to as *empirical risk minimization* (ERM) and was initially theorized by Vapnik (See [Vapnik and Chervonenkis, 1971, Vapnik, 1991]). The empirical risk $R_{\text{emp}}(h)$ of a model h is a good estimate of its true risk $R(h)$ in the sense that its expected value equals $R(h)$[1]. However, this is not enough to guarantee that, for a training dataset D, the minimizer of $R_{\text{emp}}(., D)$ has a true risk close to the minimal true risk [Vapnik, 1991]. For instance, if $\mathcal{H}$ is complex enough to contain a model h such that $h(x^\ell) = y^\ell, \ell = 1, \ldots, t$, then, while yielding zero empirical error, h might be nothing more than a memorization of the training data. Thus, h is likely to make poor predictions on unseen data, resulting in a high true risk. This *overfitting* scenario is illustrated in Figure 1.6 for a one-dimensional regression task ($\mathcal{X} = \mathbb{R}, \mathcal{Y} = \mathbb{R}$).

[1] Since (X^ℓ, Y^ℓ) are i.i.d. according to distribution $\mathcal{D}$, $\mathbb{E}[R_{\text{emp}}(h)] = \frac{1}{t}\sum_{\ell=1}^t \mathbb{E}_{(X^\ell,Y^\ell)\sim\mathcal{D}}[l(h(X_\ell), Y_\ell)] = \frac{1}{t}\sum_{\ell=1}^t R(h) = R(h)$.

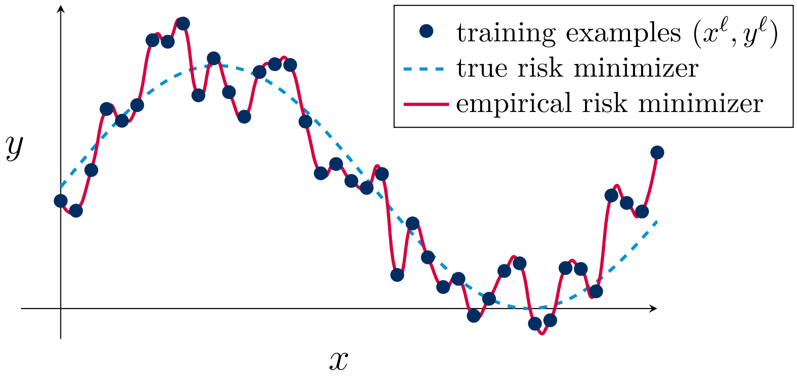


Figure 1.6: Overfitting illustration for a regression task in one dimension ($X = \mathbb{R}, \mathcal{Y} = \mathbb{R}$).

As suggested by Figure 1.6, the interpolation of all training examples is the result of a highly flexible hypothesis class containing extremely sensitive models, i.e., for which a small input change can result in a large output variation. Then, a way to overcome overfitting is to promote models with low sensitivity, by penalizing the empirical risk using *regularization* functions that measure models' sensitivity. For instance, considering linear models $h_w(x) = w_1x_1 + \ldots + w_nx_n$, a suitable regularization function is the Euclidean norm of the weight vector w defined by $\|w\|_2 = \sqrt{\sum_{i=1}^n w_i^2}$. Indeed, using Cauchy-Schwartz inequality we have that for any $x, x' \in \mathcal{X}$, $|h_w(x) - h_w(x')| \leq \|w\|_2\|x - x'\|_2$ and thus when $\|w\|_2$ is small, a minor input variation $\|x - x'\|_2$ yields a small output change $|h_w(x) - h_w(x')|$. In the following, the regularized empirical risk approach is further investigated.

3.1.2 Regularized Empirical Risk Minimization

Regularized empirical risk minimization (RERM) [Vapnik, 1995] involves minimizing the empirical risk along with a regularization term to avoid overfitting. Usually, functions h are parametrized by some parameters w and thus the hypothesis class can be defined as $\mathcal{H}_W = \{h_w | w \in W\}$, where W is the set of admissible parameters. In this context, the regularization term is a function $r : W \to \mathbb{R}$ that penalizes elements of $\mathcal{H}_W$ with a high sensitivity. A RERM can thus be defined as follows:

Definition 1.21 (RERM). *The RERM problem for hypothesis class $\mathcal{H}_W$, loss function l, regularization function r, and training data $D = \left\{\left(x^\ell, y^\ell\right)\right\}_{\ell=1}^t$, is the following optimization problem:*

$$\min_{w \in W} \frac{1}{t} \sum_{\ell=1}^t l(h_w(x^\ell), y^\ell) + \lambda r(w)$$

where $\lambda \in \mathbb{R}_+$ is a regularization hyperparameter.

Intuitively, RERM allows finding a tradeoff between fitting the training data and keeping the model simple enough to make accurate predictions on unseen data. This tradeoff is controlled by the regularization parameter λ: the higher the regularization the lower the sensitivity of the solution of the RERM problem. In the following, model's parameters w are supposed to be d-dimensional vectors, with $d \in \mathbb{N}^*$. In this case, regularization functions are typically ℓ_p-norms, defined below:

Definition 1.22 (ℓ_p-norms). *For any $p \geq 1$, the ℓ_p-norm function is denoted by $\|w\|_p$, and defined as:*

$$\|w\|_p = \left(\sum_{i=1}^{d} |w_i|^p\right)^{1/p}, \text{ for any } w \in \mathbb{R}^d.$$

Remark that the Euclidean norm corresponds to the case $p = 2$. More generally, ℓ_p-norms provide a structured way to control the model complexity by pushing the magnitude of the coefficients towards zero. The choice of ℓ_p-norms as regularization functions is further supported by their mathematical properties, which allow for *efficient optimization and theoretical analysis of the RERM optimization problems.* In particular, an important property is *convexity*, which is defined in the following paragraph:

Basic notions of convex analysis A fundamental concept of *convex analysis* [Rockafellar, 1997, Boyd and Vandenberghe, 2004, Nesterov et al., 2018] is that of *convex set.*

Definition 1.23 (convex set). *A set $W \subseteq \mathbb{R}^d$ is convex if $tw + (1-t)v \in W$, for any $v, w \in W$ and $t \in [0, 1]$.*

Let *domf* denotes the domain of a function $f : \mathbb{R}^d \to \mathbb{R}$, i.e., the set of points on which f is finite. Then, convex functions are defined as follows:

Definition 1.24 (convex function). *A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if domf is a convex set and $f(tw + (1-t)v) \leq tf(w) + (1-t)f(v)$, for any $v, w \in \mathbb{R}^d$ and $t \in [0, 1]$. A function f is said to be strictly convex if the inequality is strict whenever $w \neq v$ and $t \in]0, 1[$. Finally, f is said to be concave if $-f$ is convex.*

As illustrated in Figure 1.7 a function is convex if its *epigraph* (the region above its graph) forms a convex set (for any two points in it, the line segment connecting them stays inside the set).

Finally, *strong convexity* ensures that the gap between the line segment joining points $(w, f(w))$ and $(v, f(v))$ and the curve of f grows quadratically with the distance between u and w, i.e.,:

Definition 1.25 (strong convexity). *A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be strongly*

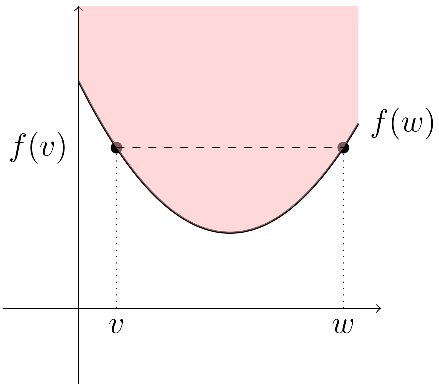


Figure 1.7: A convex function $f : \mathbb{R} \to \mathbb{R}$

convex with modulus $\alpha > 0$ *if for all* $w, v \in \mathbb{R}^d$ *and for all* $t \in [0, 1]$*, we have:*

$$f(tw + (1-t)v) \leq tf(w) + (1-t)f(v) - \frac{\alpha}{2} t(1-t)\|w - v\|^2.$$

In this case f *is said to be* α*-strongly convex*

Then, a *convex optimization problem* refers to the minimization of a convex function f over a convex set W. A key property of convex optimization problems is that any local minimum is also a global minimum, and is unique if f is strongly convex (see for instance [Boyd and Vandenberghe, 2004]- Section 4.2.2 and 9.1.2). Classic optimization algorithms are detailed in Section 3.1.4. Note that to ensure that the optimization problem is well-posed and to obtain convergence guarantees of optimization algorithms, f is often further assumed to be *proper* and *closed*, i.e.:

Definition 1.26 (proper closed convex function). *If a function* $f : \mathbb{R}^d \to \mathbb{R}$ *is convex, then it is a proper function if* $\mathrm{dom} f \neq \emptyset$ *and* f *never takes the value* $-\infty$*. Furthermore, it is a closed function if for each* $\alpha \in \mathbb{R}$ *the sublevel set* $\{w \in \mathrm{dom}\, f \mid f(w) \leq \alpha\}$ *is a closed set.*

ℓ_p-norms are convex functions, and so are standard loss functions l (e.g., squared loss, absolute loss, hinge loss). Thus, for example, using a linear model $h_w(x) = w^\top x$, function $f(w) = \frac{1}{t}\sum_{\ell=1}^{t} l(h_w(x^\ell), y^\ell)$ is a convex function, as it is a combination of a linear function and a convex function. *Therefore, for convex sets of admissible parameters* W*, RERM problems are often formulated as convex optimization problems, which enables the use of efficient optimization algorithms converging to global minima* (detailed in Section 3.1.4). Below, we present a representative example of ℓ_2-regularized empirical risk minimization problem, which will be revisited later in Chapter 3 and 4.

Linear support vector machine Introduced for binary classification (i.e., $\mathcal{Y} = \{-1, 1\}$), the linear *support vector machine* (SVM) is a well-established RERM algorithm that dates back from the 1990s [Boser et al., 1992]. It consists in learning a classifier based on a linear regression model $h_{w,b}(x) = b + w_1 x_1 + \ldots + w_d x_d$ with $W \subseteq \mathbb{R}^d$, that outputs 1 if $h_{w,b}(x) \geq 0$ and -1 otherwise. The loss function is the hinge loss, i.e., $l(h(x), y) = \max\{0, 1 - h(x)y\}$ and the regularization function is taken as $r(w) = \frac{1}{2}\|w\|_2^2$. Note that the squared ℓ_2-norm is used for simplifying the optimization by eliminating the square root (as the function $f : x \to x^2$ is strictly increasing minima are preserved). Finally, introducing positive slack variables $\epsilon_\ell = \max\{0, 1 - h(x^\ell)y^\ell\}$ modeling the error on data point $(x^\ell, y^\ell), \ell = 1, \ldots, t$, the resulting RERM optimization problem reads as follows:

$$\begin{aligned} \min_{w \in \mathbb{R}^d, b \in \mathbb{R}} \quad & \frac{1}{t} \sum_{\ell=1}^{t} \epsilon_\ell + \frac{\lambda}{2}\|w\|_2^2 \\ & y^\ell (w^\top x^\ell + b) \geq 1 - \epsilon_\ell, \ell = 1, \ldots, t \\ & \epsilon_\ell \geq 0, \quad \ell = 1, \ldots, t \end{aligned} \tag{1.10}$$

Intuitively, minimizing the hinge loss amounts to finding w such that most of the positive examples $(x^\ell, +1)$ are above the hyperplane $w^\top x + b = 1$ and most of the negative examples $(x^\ell, -1)$ are below the hyperplane $w^\top x + b = -1$. Here, the hinge loss is minimized conjointly with an ℓ_2-regularization term which allows controlling the complexity of the learned model by pushing the coefficients towards small values and thus limiting the model sensitivity. The regularization term here has a particular graphical interpretation since $r(w)$ is inversely proportional to the *margin*, that is the space in between the two hyperplanes $w^\top x + b = 1$ and $w^\top x + b = -1$, as it is illustrated in Figure 1.8 for $d = 2$. Indeed, let x be a point on the decision boundary $w^\top x + b = 0$, the distance from x to the hyperplane $w^\top x + b = +1$ is the real value $\gamma_+ \in \mathbb{R}_+$ such that $w^\top(\gamma_+ w + x) = 1$, i.e., $\gamma_+ = 1/\|w\|_2$. Summing up with the distance to the negative border $w^\top x + b = -1$, the margin, denoted by γ, equals $2/\|w\|_2$.

Therefore SVM finds the maximum margin separating hyperplane. In the next section, we present *sparsity-inducing* regularization functions, embodied by the ℓ_1-norm, that reduce model complexity by shrinking coefficients towards zero and thus provide *sparse* coefficients vectors.

3.1.3 Sparsity-inducing Regularization

A *sparsity-inducing* regularization [Tibshirani, 1996, Bach et al., 2012] refers to a regularization function that favors *sparse* model parameters (i.e., with a large proportion of zero components). Compared to other regularizations such as the ℓ_2-norm, they have

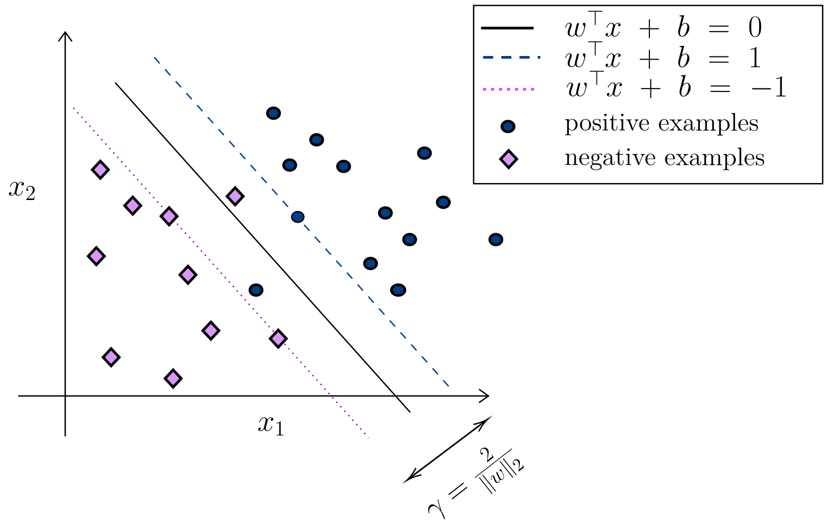


Figure 1.8: ℓ_2-regularized hinge loss (SVM) as a maximum margin optimization problem.

the added benefit of allowing the selection of the most important parameters, thus enhancing model interpretability. A first straightforward example of such sparsity-inducing regularization function is the number of non-null coefficients, called ℓ_0-norm by abuse of language, and denoted by $\|w\|_0$. However, it can easily be checked that this regularization function is non-convex and discontinuous, making its minimization very difficult from a computational point of view.

Therefore, it is common to resort to the continuous and convex, but non-smooth, ℓ_1-norm regularization ($\|w\|_1 = \sum_{i=1}^{d} |w_i|$). Indeed, as we illustrate it in the sequel, the convex and non-smooth nature of the ℓ_1-norm allows encouraging sparse solutions. Note that non-smooth is understood here as non-differentiable, i.e., there exist points where the gradient does not exist[2]. The property of *subdifferentiability*, that we briefly introduce below, generalizes differentiability for non-differentiable convex functions, and is a useful tool for their optimization:

Definition 1.27 (subdifferentiability and subgradients). *A convex function $f : \mathbb{R}^d \to \mathbb{R}$ is subdifferentiable at $w \in dom f$ if there exists $u \in \mathbb{R}^d$ such that for any $z \in dom f$, $f(z) \geq u^\top (z - w) + f(w)$. Any vector u verifying the latter inequality is a subgradient of f at w. The set of all subgradients of f at w, denoted by $\partial f(w)$, is called the subdifferential of f at w. Finally, f is said subdifferentiable if it is subdifferentiable at any $w \in dom f$.*

Remark 1.6. If f is convex, f is differentiable at $w \in dom f$ iff $\partial f(w) = \{\nabla f(w)\}$, where $\nabla f(w)$ denotes the gradient of f at w. For more in-depth results and proofs see [Bauschke

[2] For $d = 1$, this means that $\frac{f(w+h)-f(w)}{h}$ does not admit a limit when $h \to 0$. A similar formulation of non-differentiability exists for $d > 1$ (see [Bauschke and Combettes, 2011], Chapter 17).

and Combettes, 2011].

The convex univariate absolute value function $f : w \rightarrow |w|$ upon which is built the ℓ_1-norm, while not being differentiable at $w = 0$, is subdifferentiable. Indeed, as it can be graphically observed on Figure 1.9, the univariate absolute value function admits the following subdifferential:

$$\partial f(w) = \begin{cases} \{1\} & \text{if } w > 0 \\ [1, -1] & \text{if } w = 0 \\ \{-1\} & \text{if } w < 0 \end{cases} \tag{1.11}$$

Then, the absolute value function non-smoothness at $w = 0$ is embodied by an infinite number of subgradients and a kink in the curve. This kink then gives rise to angularities at points with zero coefficients in the landscape of ℓ_1-norm regularized objective functions, which, during optimization, favors solutions with exactly zero coefficients. A pioneering and emblematic example of ℓ_1-regularized RERM problem is the *LASSO* regression [Tibshirani, 1996], that we present in the following. Before that, we give a necessary and sufficient optimality condition for any convex function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ that directly follows from the definition of the subdifferential (see Definition 1.27):

$$w^* \in \underset{w \in \mathbb{R}^d}{\arg\min} f(w) \Longleftrightarrow 0 \in \partial f(w^*) \tag{1.12}$$

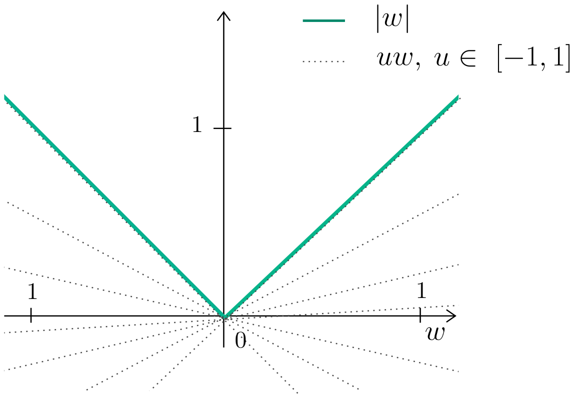


Figure 1.9: The absolute value function : $x \rightarrow |w|$ and some affine minorants at $w = 0$.

LASSO Regression LASSO (Least Absolute Shrinkage and Selection Operator) [Tibshirani, 1996, Hastie et al., 2015a] is a ℓ_1-regularized least square linear regression, i.e, a RERM problem with regression examples (i.e $\mathcal{Y} = \mathbb{R}$), linear models (i.e., $h_{w,b}(x) = b + w_1x_1 + \ldots + w_dx_d$), squared loss (i.e., $l(h(x), y) = \frac{1}{2}(h(x) - y)^2$) and ℓ_1 regularization (i.e, $r(w) = \|w\|_1$). Hence, it consists of solving the following optimization problem:

$$\min_{w \in \mathbb{R}^n, b \in \mathbb{R}} \frac{1}{2t} \sum_{\ell=1}^{t} ((w^\top x^\ell + b) - y^\ell)^2 + \lambda \|w\|_1 \tag{1.13}$$

To illustrate how ℓ_1-norm's non-differentiability yields sparse solutions for Problem 1.13, we examine the toy example of a one-dimensional LASSO regression.

Example 1.10. *Consider a one-dimensional linear model without intercept, i.e., $h_w(x_1) = w_1 x_1$. Also, let us denote $X = (x^\ell)_{\ell=1}^t$ and $Y = (y^\ell)_{\ell=1}^t$ the vectors containing the input and label data respectively, where input data is assumed to have been priorly standardized, i.e., $X^T X = 1$. Then, Problem 1.13 boils down to minimizing*

$$f(w_1) = \frac{1}{2t} \|Y - w_1 X\|_2^2 + \lambda |w_1| \tag{1.14}$$

whose subdifferential corresponds to $\{\frac{1}{t}(Y - w_1 X)X^T + \lambda s | s \in \lambda \partial |.|(w_1)\}$, for any $w_1 \in \mathbb{R}$.

In Example 1.10, combining optimality Condition 1.12 and the definition of $\partial|.|(w_1)$ (see Equation 1.11), we obtain the following optimality condition:

$$w_1^* \in \arg\min_{w \in \mathbb{R}} f(w_1) \iff \begin{cases} \frac{-1}{\lambda t}(Y - w_1^* X)X^\top = 1 & \text{if } w_1^* > 0 \\ \frac{-1}{\lambda t}(Y - w_1^* X)X^\top \in [1, -1] & \text{if } w_1^* = 0 \\ \frac{-1}{\lambda t}(Y - w_1^* X)X^\top = -1 & \text{if } w_1^* < 0 \end{cases}$$

Since $XX^T = 1$, by multiplying the right-hand equations by X^T, we finally obtain the following closed-form solution for Problem 1.13:

$$w_1^* = \text{sign}\left(\frac{YX^T}{t}\right)\left[\frac{|YX^T|}{t} - \lambda\right]_+ \tag{1.15}$$

Hence, once the regularization hyperparameter exceeds the threshold $\lambda_{lim} = \frac{|YX^T|}{t}$, w_1 is set to zero.

This behavior can be compared to that of *ridge regression* [Hoerl and Kennard, 1970] which employs squared ℓ_2-regularization instead of ℓ_1-regularization. The ridge regression objective function thus reads as $g(w_1) = \frac{1}{2t}\|Y - w_1 X\|_2^2 + \frac{\lambda}{2}\|w_2\|_2^2$. As g is differentiable with $g'(w_1) = \frac{1}{t}(Y - w_1 X)X^T + \lambda w_1$, ridge regression admits the closed-form solution $w_1^* = \frac{YX^\top}{t(1+\lambda)}$. Consequently, coefficient w_1 is not set to zero at any given time, but pushed towards zero while λ increases.

As an illustration, the LASSO objective function f (see Equation 1.14) computed with three learning examples given by $X = (0.48, 0.66, 0.58)$ and $Y = (0.43, 0.63, 0.71)$

is represented in Figure 1.10 (top) for increasing values of λ. An angularity in the curve of the objective function can be seen at $w_1 = 0$ on all graphs, and as λ increases, this sharp angle quickly attracts the minimum of the objective function. We indeed recover the fact that for $\lambda \geq \lambda_{lim} = 0.34$, $w_1 = 0$ is the optimal solution. In contrast, Figure 1.10 (bottom) that represents the objective function of the ridge regression, shows that the optimal solution of the ridge regression converges towards zero as λ increases without ever hitting exactly zero.

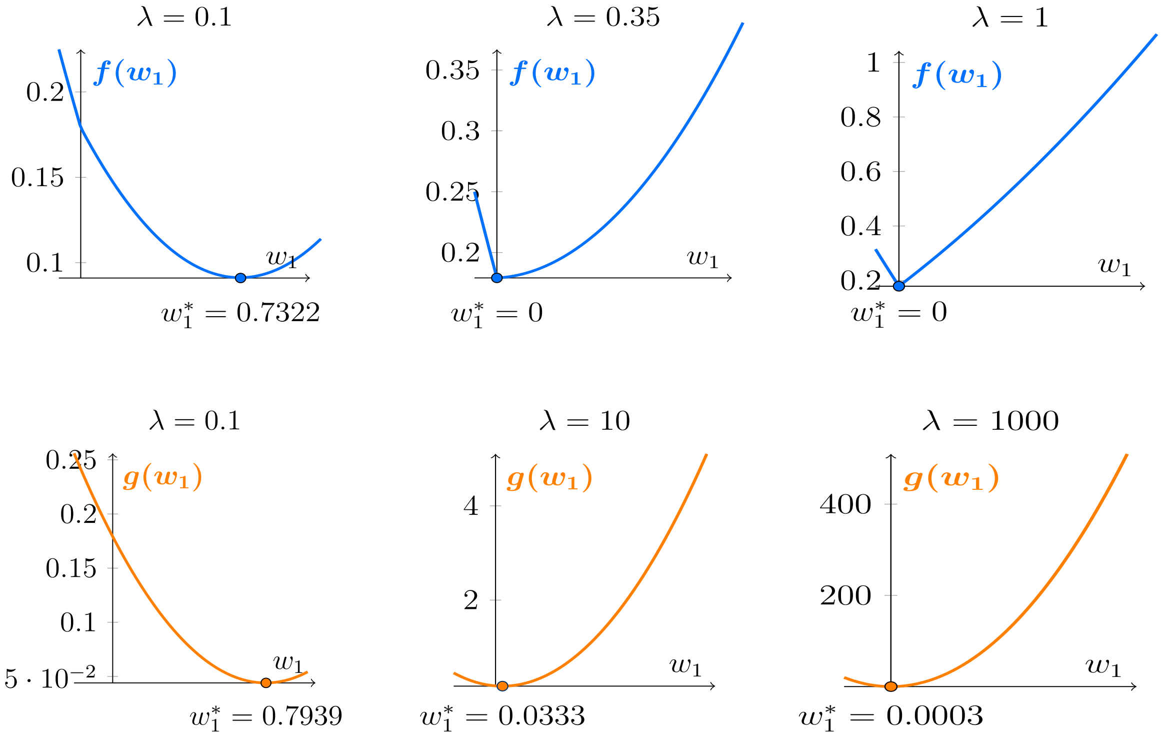


Figure 1.10: LASSO (top) and ridge (bottom) objective functions for increasing λ values.

Other examples of sparsity-inducing regularization functions are mixed norms such as the *elastic-net* [Zou and Hastie, 2005], that combines ℓ_1 and ℓ_2 regularization and thus reads as: $r(w) = \lambda\|w\|_1 + \beta\|w\|_2^2$. Also, *$\ell_1$-group norm* [Roth and Fischer, 2008, Huang and Zhang, 2010] are used to select parameters according to a group structure, i.e., coefficients belonging to a given group are put to zero all together. For instance, given a set of groups $\mathcal{G}$, a standard choice is $r(w) = \sum_{S \in \mathcal{G}} \|w_S\|_2$, where w_S is the restriction of w to the components in S. Similarly to the ℓ_1-norm, these regularization functions admit points of non-differentiability, allowing to encourage sparse solutions in the optimization process. A more in-depth analysis of this class of regularization functions is available in [Bach et al., 2012]. Finally, we can also mention other ℓ_1-norm-based penalties that encourage solutions with interesting properties, such as integer-valued coefficients [Belahcene et al., 2020] or uniform coefficients (with few distinct values) [Tibshirani et al., 2005].

3.1.4 Convex Optimization Algorithms

As previously discussed, the formulation of a supervised learning problem typically involves selecting a hypothesis class (model), a loss function, and a regularization function, thereby forming a RERM problem, which we assume to be *convex* here. The learning algorithm then consists of training the model, i.e., solving the RERM problem.

In the following, we provide a brief overview of standard optimization methods used for solving convex optimization problems, with a particular focus on RERM problems. To this end, we borrow from the books [Bubeck et al., 2015, Nesterov et al., 2018] the clarifying distinction between first and second order *black-box* convex optimization algorithms that solely rely on gradient and hessian information, and *structural* convex optimization algorithms, which additionally exploit the specific structure of the optimization problem, such as the analytical form of the objective function (e.g., linear or quadratic). For a more in-depth discussion on convex optimization algorithms, we refer the reader to these books, as well as to [Bertsekas, 2015] and [Sra et al., 2011] for machine learning-specific perspectives.

The optimization problem under consideration is therefore of the following form:

$$\min_{w \in W} f(w) := R(w) + \lambda r(w) = \frac{1}{t} \sum_{\ell=1}^{t} l(h_w(x^\ell), y^\ell) + \lambda r(w) \tag{1.16}$$

where f, R, r are convex functions and W is a convex set.

Black-box first and second-order convex optimization methods

In this section, we present iterative optimization procedures that start with an initial solution w^0 and use at each iteration k the information of the gradient at the current solution $\nabla f(w^k)$ (*first-order* methods) and possibly of the Hessian matrix $\nabla^2 f(w^k)$ (*second-order* methods) to improve the solution. Thus, unless specified, f is assumed to be twice differentiable, and with L-Lipschitz gradients, i.e., such that for any $w, v \in \mathbb{R}^d$ $\|\nabla f(w) - \nabla f(v)\|_2 \leq L\|w - v\|_2$ (except when specified, W is identified to $\mathbb{R}^d$). Finally, the different methods are compared in terms of the convergence rate of the objective function, i.e., a method is said to have a convergence rate in $O(g(k))$ when the difference $f(w^k) - f(w^*)$ is in $O(g(k))$.

First-order algorithms The basic first-order algorithm is the *gradient-descent* algorithm [Cauchy et al., 1847, Nocedal and Wright, 1999], which updates the current solution as follows:

$$w^k = w^k - \eta_k \nabla f(w^k) \tag{1.17}$$

where $\eta_k \in \mathbb{R}_+^*$ is referred to as the *step-size* or *learning rate*. This iterative procedure is known to admit a convergence rate in $O(\frac{1}{k})$, which can be improved to $O(\frac{1}{k^2})$ using a variation of the algorithm referred to as the *Nesterov's accelerated gradient-descent* [Nesterov, 1983]. When f is strongly convex the convergence rate is in $O(C^k)$, $C \in [0, 1]$. Also, it is important to note that when f is only sub-differentiable (and thus $\nabla f(w^k)$ is replaced with a subgradient $s^k \in \partial f(w^k)$), the convergence is much slower as it is in $O(\frac{1}{\sqrt{k}})$ [Nesterov et al., 2018].

As f is here the objective function of the RERM Problem 1.16, the computation of gradient $\nabla f(w^k)$ requires computing the gradient of the loss $l(h_w(x^\ell), y^\ell)$ for any example x^ℓ, y^ℓ yielding a computational cost in $O(dt)$ at each iteration. In the context of large-scale problems (a large number of variables d or a large number of training examples t) [Bennett and Parrado-Hernández, 2006, Bottou and Bousquet, 2007], such a computation may be prohibitive. Thus, alternatives to gradient descent have been proposed to alleviate the computational cost of gradient computing. Among them, *stochastic gradient-descent* (SGD) [Bottou, 2010] or *online gradient-descent* (OGD) [Shalev-Shwartz, 2012] are particularly computationally efficient because, at each iteration, they compute the gradient of the loss attached to a single example (randomly drawn in the case of SGD and lastly received in an online streaming in the case of OGD). The context of online learning has led to a distinct body of literature focusing on two optimization paradigms: *online mirror descent* (OMD) methods [Beck and Teboulle, 2003] (of which OGD is a special case) and *follow-the-regularized-leader* [Shalev-Shwartz, 2007, 2012] (also known as *regularized dual averaging* (RDA) [Xiao, 2009] for ℓ_1-regularized loss), both of which will be presented in more details in *Chapter 6*, which addresses the online learning of preference models.

When Problem 1.16 is constrained, i.e., $W \neq \mathbb{R}^d$, the scheme given by Equation 1.17 can be extended to *projected gradient-descent*, where at each iteration the updated point w^k is projected back onto the feasible set. We can also mention the *conditional gradient-descent* (also known as the *Frank-Wolfe algorithm*) [Frank et al., 1956, Jaggi, 2013] that does not rely on projections but rather linearizes f at the current solution at each iteration and performs linear programming (LP) over W, yielding a low computational complexity per iteration when LP over W reduces to a simple problem.

Second-order algorithms Compared to first-order optimization methods, second-order methods offer faster convergence due to the inclusion of curvature information carried by the Hessian matrix $\nabla^2 f(w^k)$. The basic second-order algorithm is the Newton's method [Kantorovich, 1949, Nocedal and Wright, 1999], which updates the current solution as follows:

$$w^{k+1} = w^k - \left[\nabla^2 f\left(w^k\right)\right]^{-1} \nabla f\left(w^k\right) \tag{1.18}$$

When the algorithm is initialized with a starting point w^0 sufficiently close to the optimal solution w^* of Problem 1.16, and f is strongly convex, it converges to a solution satisfying $f(w^*) - f(w^k) \leq \epsilon$ in $O(\log(\log(1/\epsilon)))$ iterations, achieving faster convergence than gradient descent-based methods (see, for instance, Chapter 1 of [Nesterov et al., 2018]). However, storing the Hessian matrix and performing operations on it (notably inversion) pose computational challenges. These difficulties can be addressed using a *quasi-Newton* method, where the Hessian matrix is not explicitly computed but approximated [Goldfarb, 1970, Nocedal, 1980]. Similarly to gradient descent, stochastic and online variants of quasi-Newton methods have been proposed to handle large-scale learning problems [Schraudolph et al., 2007, Byrd et al., 2016, Sun et al., 2019].

Structural convex optimization methods

We now discuss optimization methods that leverage the specific structure of Problem 1.16, to achieve faster convergence (potentially with a higher computational cost per iteration).

Optimization methods for sparsity-inducing regularization In the context of sparse learning for instance, there is a focus on optimization methods specifically suited for the case where $r(w)$ is a sparsity-inducing penalty such as the ℓ_1-norm. Complete descriptions of such methods can be found in [Bach et al., 2012] or [Hastie et al., 2015b]. Among them, *proximal methods* leverage the division of the objective function in a differentiable part (the loss, i.e., $R(w)$ in Problem 1.16) and a non-differentiable part (the sparsity-inducing penalty, i.e., $r(w)$ in Problem 1.16). More precisely, at each iteration $R(w)$ is linearized around the current solution using its gradient and a *proximal term* is added to maintain the next solution in a neighborhood of the current solution, i.e., for some $L > 0$:

$$w^{k+1} = \underset{w \in \mathbb{R}^d}{\arg\min}\, R(w^k) + \nabla R(w^k)^\top (w - w^k) + \lambda r(w) + \frac{L}{2}\left\|w - w^k\right\|_2^2 \tag{1.19}$$

Such algorithm is known to admit a convergence rate in $O(\frac{1}{k})$, which can be improved to $O(\frac{1}{k^2})$ using a variant that is referred to as the *fast iterative shrinkage-thresholding* (FISTA) algorithm [Beck and Teboulle, 2009]. Thus, it provides faster convergence than gradient-descent with non-differentiable functions. This is obtained without increasing the computational cost of each iteration, as for instance when r is the ℓ_1-norm, Problem 1.19 admits a closed-form solution that can be computed in $O(dt)$. *Chapter 6* will revisit this type of optimization method in the context of online learning for preference models. Notably, the proof of the closed-form solution of Problem 1.19 for the ℓ_1-norm is given in Lemma 2 of the Appendix C).

An alternative type of method, referred to as *coordinate descent* methods, exploit the additive decomposition of the ℓ_1 regularization (which is the sum of the variables's absolute values), and optimize w.r.t. one variable only at each iteration [Wu and Lange, 2008, Friedman et al., 2010]. Finally, ℓ_1-regularized loss functions can be minimized using *iteratively re-weighted least squares* (IRLS) methods that leverage links between ℓ_1-regularization and l_2-regularization to solve a sequence of l_2-regularized problems [Daubechies et al., 2010]. Further details are provided in *Chapter 3*, where this type of optimization method will be studied for learning preference models.

Alternating direction method of multipliers (ADMM) When the model parameter are constrained, i.e., $W \neq \mathbb{R}^d$, the computational efficiency of the previous methods might be lost. A widely-used approach to circumvent this issue is to use *alternating direction method of multipliers* (ADMM) methods [Glowinski and Marroco, 1975, Boyd et al., 2011] that exploit constraint structure to break down complex optimization problems into smaller subproblems, reducing computational complexity. A presentation of this method is given in *Chapter 6*, where it is explored for learning preference models with constraints on the parameters, in the online setting.

Interior-point methods (IPM) Alternatively, Problem 1.16 can sometimes be classified into typical problem categories, such as *linear programming* (LP) (linear objective and linear constraints), *quadratic programming* (QP) (quadratic objective and linear constraints) *second-order cone programming* (SOCP) (linear objective and second-order cone constraints [Lobo et al., 1998] which includes quadratically constrained quadratic program (QCQP)), or *semi-definite programming* (SDP) (linear objective and affine combination of symmetric matrices constrained to be positive semi-definite [Vandenberghe and Boyd, 1996]). For instance, the LASSO regression and SVM optimization problems (Problem (1.13) and Problem (1.10)) can be formulated as QP problems.

LP, QP, SOCP, and SDP problems can be solved with high precision by standard numerical solvers that use *interior points methods* (IPM) (or *active-set methods* that we describe in the next paragraph). As such solvers are used in this thesis to perform LP, QP, and QCQP optimization tasks (see *Chapter 2-4*), we give a brief description of IPM below.

IPM, originally proposed for LP [Karmarkar, 1984, Renegar, 1988] (see [Nesterov and Nemirovskii, 1994] for a summary of historical contributions), works by starting from a point inside the feasible region and iterating towards the optimal solution, using a barrier function that maintains the iterate within the feasible region. More precisely, it starts with the following reformulation of Problem 1.16:

$$\begin{aligned} &\min c^\top v \\ &\text{s.t. } v \in \mathcal{V} \end{aligned} \tag{1.20}$$

where $(c, v) \in \mathbb{R}^{d+1}$ and $\mathcal{V} \subseteq \mathbb{R}^{d+1}$ is a convex set[3].

Then, IPM leverages a *barrier* function, i.e., a smooth and strongly convex function $B : \text{int}(\mathcal{V}) \to \mathbb{R}$ such that $B(v) \xrightarrow[v \to Fr(\mathcal{V})]{} +\infty$, where $Fr(\mathcal{V})$ is the frontier of $\mathcal{V}$ and $\text{int}(\mathcal{V})$ its interior. Then, the following approximating problem is considered for any $t > 0$:

$$\min F_t(v) = tc^\top v + B(v) \tag{1.21}$$

Under mild assumptions, the sequence of solutions $v(t) \in \text{int}(\mathcal{V})$ obtained by solving Problem 1.21 with increasing values of t is such that $v(t) \xrightarrow[t \to \infty]{} v^*$, where v^* is the optimal solution of Problem 1.20 [Nemirovski and Todd, 2008, Nesterov et al., 2018]. Rapid convergence can be guaranteed by solving Problem 1.21 for a sequence of increasing values t_i, where at each iteration a Newton's algorithm (see Equation 1.18) is launched with an initial solution taken as the previous solution $w(t_{i-1})$. However, this requires having a barrier function B whose gradient and Hessian matrix are well-defined and easily computable. This is the case for instance for LP, i.e., when $\mathcal{V} = \{v \in \mathbb{R}^{d+1} | a_i v - b_i \leq 0, i = 1, \ldots, m\}$. In this case, using the *logarithmic-barrier* $B(v) = -\sum_{i=1}^{m} \log\left(a_i^\top v - b_i\right)$, it is known that the IPM algorithm admits a convergence rate in $O(m \exp \frac{-k}{\sqrt{m}})$ (see for instance [Nesterov et al., 2018, Bubeck et al., 2015] both in Chapter 5). Note that this rate depends on m the number of constraints, and furthermore, each iteration requires performing Newton's steps. Thus, such an optimization method can be prohibitive when the number of variables d or constraints m is large.

Active-set methods Standard numerical solvers also widely rely on *active-set* methods. These methods, which include the *simplex* algorithm [Dantzig, 1951] for LP, work by iteratively selecting a set of inequality constraints potentially active (i.e., for which equality holds) at the optimum, and solving a smaller optimization problem only using these constraints. This principle has also been used in machine learning to solve large-scale learning problems, notably for solving SVM (see Problem 1.10) with a large number of examples, using the *sequential minimal optimization* (SMO) algorithm [Platt, 1998, Fan et al., 2005].

[3] it can easily be checked that Problem 1.16 is equivalent to $\min_{(t,w) \in \mathbb{R} \times W \text{ s.t. } t \geq f(w)} \quad t$

3.2 Learning Preference Models

3.2.1 Preference Learning Singularities and Challenges

Designing supervised learning algorithms for the preference models introduced in Section 1 requires taking into account the specific structure of preference data and models.

Training data As it is standard in preference elicitation (see Section 2), preference information may be available in the form of utility values (ratings/global evaluations) or rankings of alternatives, the latter being described by the vector of their consequences $x = (x_1, \ldots, x_n)$ w.r.t. n viewpoints (where x_i may directly represents the marginal utility w.r.t. the i^{th} viewpoint; see Section 1).

In the case where preference information is available in the form of utility values, the learning task boils down to a regression task where the targets y^ℓ, $\ell = 1, \ldots, t$ are the alternatives' utilities. When the utility information is ordinal, i.e., $y^\ell \in \{1, \ldots, K\}$, it forms a multi-class classification problem with ordered class labels (also known as *ordinal regression*). A simple example is the classification of scientific papers in the ordinal categories $\{1 : \text{reject},\ 2 : \text{weak reject},\ 3 : \text{weak accept},\ 4 : \text{accept}\}$. This case is also known as multipartite ranking (or bipartite when $K = 2$). However, utility numerical information is considered difficult to obtain from the DM, who may not be able to quantify the intrinsic utility of an alternative. On the other hand, preference information in the form of pairwise comparisons (rankings of pairs) such as $x^\ell \succsim x'^\ell$ (alternative x^ℓ is preferred to alternative x'^ℓ), is considered easier to acquire. Therefore, the training dataset is often encountered as a collection of pairwise preference examples $D = \{(x^\ell, x'^\ell)\}_{\ell=1}^t$, where $x^\ell \succsim x'^\ell$ for instance. Note that rankings of more than two alternatives could also be considered.

These preference databases can be derived from individual questionnaires, as is traditionally the case in preference elicitation, but they can also be collected from the observation of behavior on the web (e.g., search engines, social medias, streaming platforms). For instance, one can think of restaurant/movie choices or ratings, or evaluations of answers provided by an AI-powered chatbot. A standard reference for preference data is the website https://preflib.simonrey.fr/.

Remark 1.7 (preference learning as binary classification). Preference learning from pairwise comparisons can be viewed as a binary classification task with an antisymmetric classifier $H(x, x') = \text{sign}(h(x) - h(x')), h : \mathbb{R}^n \to \mathbb{R}$, where the inputs are of the form (x^ℓ, x'^ℓ) associated with the label +1 when $x^\ell \succ x'^\ell$ and -1 otherwise.

Loss function The employed loss function naturally has to be tailored to the preference data structure. When overall evaluations of alternatives are available, standard regression loss functions such as the squared or the absolute loss are perfectly suitable. When training data are available in the form of pairwise preference examples, the loss function is designed to penalize preference violations. Violations can be measured in a binary manner, i.e., we can define a loss function $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ such that for any preference example $x \succ x'$, and any utility function $h : \mathbb{R}^n \to \mathbb{R}$:

$$l(h(x), h(x')) = \begin{cases} 1 \text{ if } h(x') \geq h(x), \\ 0 \text{ otherwise.} \end{cases}$$

Note that it corresponds to the 0-1 loss for the binary classifier $H(x, x') = \text{sign}(h(x) - h(x'))$ (see Remark 1.7). As such loss is non-convex and discontinuous, it is very challenging to optimize. Then, it is common to resort to the continuous and convex following loss, penalizing preference violation intensity:

$$l(h(x), h(x')) = \begin{cases} \delta - (h(x) - h(x')) & \text{if } h(x') \geq h(x) - \delta, \\ 0 & \text{otherwise.} \end{cases}$$

where $\delta \geq 0$ is a threshold used to separate strict preference from indifference situations. Note that this loss corresponds to the hinge loss (for $\delta = 1$) for the binary classifier $H(x, x') = \text{sign}(h(x) - h(x'))$ (see Remark 1.7). Generalizing this idea to handle preference and indifference examples, we can define the following convex loss, referred to as the *pref-hinge loss* in this manuscript:

Definition 1.28 (pref-hinge loss). *Let P and I be the set of indices of the examples of preference and indifference respectively. The pref-hinge loss is a function $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ such that for any utility function $h : \mathbb{R}^n \to \mathbb{R}$, $\delta \geq 0$, and any example (x^ℓ, x'^ℓ):*

$$l(h(x^\ell), h(x'^\ell)) = \begin{cases} \max\{0, \delta - (h(x^\ell) - h(x'^\ell))\} & \text{if } \ell \in P \\ \max\{0, \delta - |h(x^\ell) - h(x'^\ell)|\} & \text{if } \ell \in I \end{cases}$$

Remark 1.8 (semi-orders). Parameter δ conveys the idea that the observed preferences do not necessarily form a weak order $\succsim$ (i.e., such that there exists a utility function h satisfying $h(x) > h(x')$ iff $x \succ x'$ and $h(x) > h(x')$ iff $x \sim x'$; see Remark 1.1). Preferences may indeed form a *semi-order* [Luce, 1956, Pirlot and Vincke, 2013], characterized by the existence of a utility function h and an indifference threshold $\delta \geq 0$ such that:

$$x \succ x' \iff h(x) > h(x') + \delta$$
$$x \sim x' \iff |h(x) - h(x')| \leq \delta$$

Challenges of learning decision-theoretic preference models with interactions As discussed in Section 1, preference models with interactions from decision theory (e.g., totally decomposable models with capacity-based aggregation functions, GAI-decomposable utility functions) make it possible to capture complex preference behaviors, notably overcoming the descriptive limitations of additive utility models. However, this descriptive power is achieved using a *large number of parameters*: one capacity weight $w(S)$ for each possible subset of interacting viewpoints $S \subseteq N$ in models based on a capacity w, or one sub-utility factor u_S for any $S \subseteq N$ for GAI-decomposable utility functions.

Moreover, while demonstrating strong descriptive power, these models satisfy mathematical properties that ensure some consistency and rationality in preferences. For instance, totally decomposable models are monotonic (see Definition 1.6) and thus guarantee that an alternative that is at least as good as another from every point of view cannot be judged less attractive overall. Another example is the fact the Choquet integral with a supermodular capacity (see Definition 1.13) allows modeling the preference for balanced solutions in contexts where fairness is desirable (see Example 1.5, or more generally [Lesca and Perny, 2010]). However, these desirable behaviors are guaranteed through *combinatorial sets of structural constraints* (i.e., monotonicity w.r.t. set inclusion and supermodularity) on the capacity.

Therefore, specific optimization methods able to handle such a number of parameters and constraints are needed to solve the learning problems. Additionally, the exponential number of coefficients defining the capacity entails a high risk of overfitting the training data that has to be controlled through regularization. Due to their ability to select important parameters, *sparsity-inducing regularizations* emerge as promising candidates for *controlling capacities' flexibility and obtaining a clear representation of the most significant interactions between viewpoints*. Therefore, the computational difficulties due to the large number of constraints and parameters are combined with the need for sparsity, which requires resorting to optimization techniques capable of exploiting the particular structure of large-scale ℓ_1-regularized RERM problems.

Among the totally decomposable models with capacity-based aggregation functions presented in Section 1, this thesis primarily focuses on the *Choquet integral* and the *multilinear utility*. These models, as shown by Equation 1.5 and Equation 1.9, have the advantage of being *linear in the capacity*, when the marginal utilities are known.

As a consequence, the learning of these models can generally be formulated as linear or quadratic optimization problems, depending on the selected loss and regularization functions. Indeed let us take $h_w := C_w$ or $h_w := \mathrm{ML}_w$, and $w \in W$ where W is the set of capacity. Thus, a RERM problem for learning w with a set of preference examples $D = \{(x^\ell, x'^\ell)\}_{\ell=1}^t$, where $x^\ell \succsim x'^\ell$, the pref-hinge loss and a regularization function $r : W \to \mathbb{R}$, reads as:

$$\begin{aligned}
& \min_{w\in W} \frac{1}{t}\sum_{\ell=1}^{t} l(h_w(x^\ell), h_w(x'^\ell)) + \lambda r(w) \\
:= \quad & \min_{w\in\mathbb{R}^{2^n-1}} \frac{1}{t}\sum_{\ell=1}^{t} \epsilon_\ell + \lambda r(w) \\
\text{s.t.} \quad & h_w(x^\ell) - h_w(x'^\ell) + \epsilon_\ell \geq \delta, \quad \ell = 1,\ldots,t \\
& w(S) \geq w(T), \quad S \subseteq T, \quad T, S \subseteq N \qquad (1.22) \\
& w(N) = 1 \qquad (1.23) \\
& \epsilon_\ell \geq 0, \quad \ell = 1,\ldots,t
\end{aligned}$$

where the second formulation is obtained by introducing positive error variables $\epsilon_\ell = \max\{0, \delta- \ (h(x^\ell) - h(x'^\ell))\}$ and by making explicit the constraints that define the set of capacities, namely the normalization (Constraint 1.23) and monotonicity (Constraint 1.22) constraints. Thus, by linearity of h_w w.r.t. w, such optimization problem falls into linear programming when r is the ℓ_1-norm (provided r is linearized; see Remark 2.2 in Chapter 2 for details on how the ℓ_1-norm can be linearized), and quadratic programming when r is the ℓ_2-norm. The optimization approaches developed in this thesis to solve the RERM problem make use of this particular property, and are thus, in general, not suitable for models that are non-linear in the capacity, such as the Sugeno integral (see Definition 1.17)

3.2.2 Existing Approaches for Learning Decision-theoretic Preference Models with Interactions

Totally decomposable utility functions introduced in Section 1, and in particular Choquet integrals, have been the focus of many contributions in preference learning, thus complementing the preference elicitation methods already presented in Section 2.2.2. A first approach was proposed for learning binary classifiers based on Choquet and Sugeno integrals [Grabisch and Nicolas, 1994]. Then, ordinal regression with the Choquet integral was formulated as the minimization of the pref-hinge loss over all possible pairs of alternatives included in the training set [Tehrani et al., 2012c]. Alternatively, ordinal regression with the Choquet integral can be addressed by extending logistic regression (a RERM problem based on a linear classifier and the logistic loss). Such extension is referred to as the ordinal *Choquistic regression* [Tehrani et al., 2012a, Tehrani and Hüllermeier, 2013]. The RERM learning problems presented in Section 3 such as SVM and ridge regression can be similarly extended for learning Choquet integrals in the binary classification [Tehrani, 2021] and regression setting [Kakula et al., 2020a]. Also, an efficient algorithm for learning Choquet integrals in the regression setting has been proposed [Beliakov and Wu, 2019b]. It is based on the concept of k-*interactivity* which, like k-additivity, reduces the number of variables defining the capacity, but also reduces the number of monotonicity constraints. Finally, we can mention some contributions on the learning of the multilinear model [Pelegrina et al., 2020a] and the Sugeno integral [Gagolewski et al., 2019a, Abbaszadeh and Hüllermeier, 2020, Beliakov et al., 2020]. As the Sugeno integral is not linear in the capacity, optimization methods for solving learning problems usually differ from the ones used for learning Choquet integral or multilinear models, and use for instance heuristics such as genetic algorithms [Combarro and Miranda, 2006].

The above contributions solely address the issue of learning the capacity parametrizing the Choquet integral, the multilinear model or the Sugeno integral, assuming that marginal utilities are known. Other contributions tackle the problem of learning both types of parameters simultaneously. Beyond the methods proposed in the preference elicitation literature (see Remark 1.5), we can mention the *Choquistic utilitarian regression* [Tehrani et al., 2014a], which extends Choquistic regression by modeling marginal utility functions as linear combinations of sigmoid functions. This same model for marginal utilities is used within a neural architecture for learning *hierarchical Choquet integrals* (i.e., extensions of the Choquet integral that rely on a hierarchy of viewpoints and aggregate marginal utilities using a distinct Choquet integral at each level of the hierarchy) [Bresson et al., 2021, Bresson, 2022]. This Choquet-based neural architecture has been leveraged for providing a post-hoc explanation of a deep neural network [Atienza et al., 2024].

All the above-mentioned contributions alleviate the computational difficulty of learning capacity-based aggregation functions by resorting in practice to prior reductions of the parameter space using k-additive capacities [Tehrani et al., 2012c, 2014a, Pelegrina et al., 2020a] (or similar restrictions such as k-interactive capacities [Beliakov and Wu, 2019b]), relaxing monotonicity constraints [Tehrani, 2021, Kakula et al., 2020a] or using hierarchical Choquet integral with predefined hierarchy [Bresson et al., 2021, Atienza et al., 2024]. Note that k-additivity can also be seen as a way to control the model's flexibility and thus prevent overfitting.

On the other hand, several attempts to control the complexity of the capacity using sparsity-inducing regularizations have been made. For instance, the ℓ_1-penalty was applied to the capacity [Anderson et al., 2014, Adeyeba et al., 2015, Kakula et al., 2020b] and the ℓ_0-penalty on the Shapley values (aggregation of Möbius masses reflecting the overall importance of each viewpoint) is considered in [Pinar et al., 2017]. The ℓ_1-penalty was also applied to the interaction indices representation [de Oliveira et al., 2022]. Thus, it appears that the choice of the capacity representation on which sparse regularization should be applied is not straightforward. Notably, the Möbius representation seems to have been little exploited for learning sparse representations of capacities, although it is particularly suitable, as will be highlighted in Chapter 2 of this thesis. Moreover, the computational challenge arising from the relaxation of k-additivity constraints is not truly addressed, as the methods are tested on small-scale problems. (less than 5 viewpoints) [Anderson et al., 2014, Adeyeba et al., 2015, Pinar et al., 2017, de Oliveira et al., 2022]. While [Kakula et al., 2020b] partially addresses the computational challenge by using online gradient descent to solve the learning problem, it does not guarantee monotonicity of the learned preference models.

Concerning the learning of GAI utility functions, as far as we know, beyond the elicitation procedures mentioned in Section 2.3, only one contribution [Bigot et al., 2012] has been proposed in the preference learning literature. This approach allows learning simultaneously the decomposition and the sub-utility functions. However, it is formulated for Boolean attributes and interactions are limited to subsets of bounded size k ($k = 2$ is used in practice) in the spirit of k-additivity. Hence, as with totally decomposable models, there is a necessity to develop computationally efficient learning techniques that yield simple decomposition with few factors, without enforcing prior restrictions on the size of interactions.

4 Conclusion

In this chapter, we first introduced key examples of preference models based on a utility function that assigns an overall value to each alternative. In particular, we introduced two classes of decomposable utility functions of increasing generality: totally and GAI-decomposable utility functions. In the first class, utility functions decompose into a set of marginal utility functions defined for each viewpoint, and an aggregation function that combines these marginal utilities into an overall value using information regarding the importance of viewpoints or groups of viewpoints. Such information can be encoded into a capacity that assigns a weight to any viewpoint group, possibly taking into account interaction effects within the groups. We have seen that taking into account these interactions between viewpoints through capacity-based aggregation functions, such as the Choquet integral or the multilinear model, makes it possible to overcome the descriptive limitations of the weighted sum and allows for modeling sophisticated decision-making behaviors. Both the Choquet integral and the multilinear model belong to the broader class of GAI-decomposable utility functions, which additively decompose into sub-utility functions generalizing the viewpoints' marginal utilities to groups of interacting viewpoints. Overall, totally and GAI-decomposable utility functions, while allowing for decomposition and therefore simple preference representations, enable the modeling of complex but natural preferences, particularly by accounting for interactions between viewpoints.

In the second section, we provided a brief overview of preference elicitation methods aimed at determining the parameters of these utility functions in close collaboration with the DM using questionnaires. We distinguished between two main approaches: decision-focused elicitation methods that use the smallest number of queries possible to obtain sufficient information on the parameters to solve a given decision problem, and approaches aimed at determining a specific parameterization that accurately represents the decision maker's preferences, generally using larger databases of preference examples. The latter type of approach naturally leads to preference learning which we introduced in the third section from the perspective of supervised learning, a general framework for learning models' parameters using labeled examples. To address the learning task, one major approach is to minimize a loss function that measures how well the model fits the examples and employing a regularization function to avoid overfitting and promote simpler models. Regularization functions of interest include sparsity-inducing regularizations, as they allow for the selection of important parameters and improve the model's interpretability. Then, convex optimization algorithms can be leveraged to solve the regularized learning problem. This general framework can be specified for preference learning using loss functions measuring preference violations on databases of preference examples, which typically take the form of pairwise comparison examples.

Preference learning has been widely used to learn decomposable utility functions. However, the proposed approaches do not overcome the limitations of the preference elicitation methods outlined at the end of Section 2. In particular, there remains a need for methods capable of learning sparse representations of preferences in a computationally efficient manner, without relying on cardinal-based prior restrictions such as k-additivity or on relaxations of the constraints that ensure the rationality of preferences (e.g., monotonicity of the preference model).

The aim of this thesis is to further explore the preference learning framework to develop computationally efficient algorithms for obtaining simple and interpretable preference representations with totally and GAI-decomposable utility functions. In particular, we aim to formulate learning problems within the RERM framework with sparsity-inducing regularization and to design computationally efficient optimization algorithms to solve these problems, leveraging the extensive literature on optimization for regularized, constrained, and large-scale learning problems. Furthermore, in doing so, we seek to provide the machine learning community with theoretically grounded and interpretable preference models. We also aim to design learning algorithms suited to contexts that extend beyond passive learning (i.e., using a pre-acquired database of preference statements). For instance, we will seek to handle incoming streams of preference examples in an online setting or to interact with the decision maker by selecting preference queries in an active learning framework. To guide the reader, Table 1.6 presents the preference models and learning parameters considered in each chapter, along with the optimization methods used.

Learning Setting	**Passive**	**Active**	**Online**
Preference Model			
Marginal utilities (with Choquet integral)		*Chapter 2*: LP	
Capacity-based aggregation function linear in the capacity	*Chapter 2*: LP, QP *Chapter 3*: IRLS, QP	*Chapter 5*: exhaustive search	*Chapter 6*: RDA,ADMM
Aggregation function non linear in its parameter		*Chapter 5*: ex. search	
GAI-decomposable utility	*Chapter 4*: SOCP		

Table 1.6: Correspondence between models, settings, chapters and optimization tools.

Chapter 2
Learning Sparse Preference Representations based on Choquet Integrals

Contents

Summary

In this chapter, we address the challenge of fitting the parameters of the preference model to the DM value system to explain or predict her preferences and propose a methodology dedicated to the identification of marginal utilities and capacities in preference models involving Choquet integrals. In particular, the objective is to derive sparse representations of capacities that do not rely on cardinal-based predetermined sparsity patterns, such as k-additivity, but reveal from preference data the most significant interactions between viewpoints. We show that we can successively learn marginal utilities from properly chosen preference examples, and sparse representations of capacities. Specifically, we propose a sparse learning approach based on *adaptive ℓ_1-regularization* for determining a sparse Möbius representation of the capacity fitted to the observed preferences. We present numerical tests to compare different regularization methods. We also show the advantages of our approach compared to basic methods that do not seek sparsity or that force sparsity a priori by requiring k-additivity. This chapter is based on several publications: [Herin et al., 2022a] for decision-making under uncertainty and [Herin et al., 2022b, 2024c] for multi-criteria/attribute decision-making.

Introduction

Due to its high versatility, the *Choquet integral* is often regarded as one of the most general compromise aggregator; it indeed encompasses various simpler aggregators like the weighted sum, OWA, and WOWA (see Subsection 1.3.1 of Chapter 1) and thus includes a rich family of aggregation functions. The Choquet integral therefore provides a natural setting to study how model complexity can be fitted to the preference system we want to describe or implement. For this reason, we focus in this chapter on the learning of the preference model that is the *Choquet integral of marginal utilities* (CIU).

The CIU model is based on two types of preference parameters: univariate *marginal utilities* defining the attractiveness of consequences on every relevant viewpoint and a set function named *capacity*, monotonic w.r.t. set inclusion, assigning a weight to every subset of viewpoints. This weighting system is then employed by the Choquet integral to perform a kind of sophisticated weighted average of the marginal utilities. In this chapter, we also consider the *bipolar Choquet integral of marginal utilities* (bi-CIU) which is an extension of CIU that allows modeling distinct behaviors at the aggregation stage when facing "good" or "bad" consequences. For this, bi-CIU uses two capacities that cooperate in weighting viewpoints or subsets of viewpoints; one applies to the positive part of the marginal utility vector whereas the other applies to the negative part.

The use of possibly non-additive capacities in CIU (resp. bi-CIU) requires the definition of 2^n (resp. 2^{n+1}) weighting parameters if n is the number of viewpoints under consideration, i.e., one (resp. two) weight for every subset of viewpoints. The multiplicity of these parameters obviously induces a significant gain of expressiveness compared to simpler preference models such as the weighted sum of marginal utilities. However, it also obviously raises the question of the parsimonious learning of the parameters defining the capacity, which could indeed prevent over-fitting of preference data and lead to more compact and more explainable preference models.

The question of learning what are the most significant subsets of interacting viewpoints and how a *sparse representation of the capacity* can be derived from preference examples remains underexplored in the literature. Some attempts to control the complexity of the capacity have been made using sparsity-inducing regularizations (see Subsection 3.2.2 of Chapter 1) on the capacity itself or on transforms of the capacities such as the Shapley values and the interaction indices (see Equation 1.7 and 1.6) [Anderson et al., 2014, Adeyeba et al., 2015, Kakula et al., 2020b, Pinar et al., 2017, de Oliveira et al., 2022]. However, seeking sparse capacities is somewhat at odds with *monotonicity*, as the capacity weights increase with set inclusion. Therefore, it is important to *discuss the appropriate capacity transformation to obtain sparse representations of the capacity.*

Notably, the *Möbius representation* seems to have been little exploited, or combined with predetermined cardinal-based sparsity patterns [Tehrani and Hüllermeier, 2013, Tehrani et al., 2012a] such as k-additivity (see Definition 1.10), which involve drastic model reductions that may significantly impact our ability to fit preference data with relevant CIU models. Additionally, as far as we know, the question of assessing *the quality of the interaction selection* in the CIU model obtained with standard sparsity-inducing regularizations such as ℓ_1-regularization has never been addressed.

Moreover, the above-mentioned contributions do not address the challenge due to the *interplay of marginal utilities and capacities* in the computation of CIU values, making the learning of these two types of parameters interdependent. This *double learning task* is all the more difficult as marginal utilities and capacities are usually not directly observable and must be derived from preference statements (comparisons of alternatives or possibly overall evaluations of alternatives, i.e., overall utility values). In practice, marginal utilities are standardly assumed to be known, or priorly acquired using elicitation procedures such as the MACBETH method [Bana e Costa and Vansnick, 1997]. However, this latter method relies on direct queries of utility values $u_i(x_i)$ for some consequences x_i, which may be difficult to answer. Other approaches under the form of *standard sequences* have been proposed [Wakker and Deneffe, 1996, Abdellaoui, 2000] (known as the *tradeoff methods*) in decision-making under risk for specific instances of the CIU model, i.e., the RDU (rank dependent utility) model [Quiggin, 2012] and the CPT (cumulative prospect theory) model [Kahneman and Tversky, 1979]. However, in addition to being formulated for particular CIU instances and decision contexts, these approaches are sensitive to the propagation of response errors along the elicitation process [Blavatskyy, 2006] (see Subsection 2.1 for a more in-depth review of related work on marginal utilities elicitation).

The double learning task has been addressed from the perspective of preference learning in the context of logistic regression (*Choquistic regression*) [Tehrani et al., 2014a] and hierarchical Choquet integrals [Bresson, 2022] by simultaneously learning both type of parameters. More precisely, these methods rely on a non-convex RERM problem (see Definition 1.21) whose variables include both the marginal utilities and the capacity, and whose resolution is based either on the use of a black-box nonlinear optimization solver, which entails a high computational cost according to the authors (tested with 2-additive capacities), or on a neural network architecture designed according to a predefined criterion aggregation hierarchy. Here, we aim to exploit the elicitation expertise developed in decision theory, and in particular, to leverage preference queries inspired by the *tradeoff method*, which allow for *isolating the respective effect of marginal utilities and capacities*. This then makes possible a sequential approach in which marginal utilities are learned first, followed by the learning of a sparse representation of capacities through the formulation of a convex RERM problem with sparsity-inducing regularizations.

Contributions and Chapter Organization First, in Section 1, we present *an approach to learn marginal utilities* in the context of *decision-making under uncertainty* where a single marginal utility needs to be defined since alternatives are acts with n possible consequences, all expressed in terms of payoffs. Then, the method is extended to the context of *multi-criteria/attribute decision-making.* In both contexts, we first formulate preference queries inspired by the tradeoff method, whose answers yield a set of linear constraints on the marginal utilities. We then formulate the learning problem as a *spline regression* where the objective is to minimize the violation of the constraints. Then, in Section 2, we propose *an approach to learn in a second step sparse representations of capacities* in the CIU and the bi-CIU model. More specifically, we start by motivating the choice of the *Möbius transform* over the capacity or the interaction index transform for applying sparsity-inducing penalties, and then formulate a ℓ_1-regularized learning problem. Limitations of this regularization in properly selecting interactions within the CIU model are then identified by leveraging the variable selection properties of ℓ_1-regularized regressions established in the statistical learning literature. We then propose addressing these issues using *adaptive ℓ_1-regularization.* Finally, in Section 3, we present numerical tests to compare the performances of our learning approach compared to baseline methods.

Notations and Preliminaries

In this chapter, the notations used are those of Section 1 of Chapter 1. More precisely, we consider a set $N = \{1, \ldots, n\}$ of n viewpoints w.r.t. which alternatives are evaluated. Alternatives are then described by vectors $x = (x_1, \ldots, x_n)$ whose components x_i are their consequences w.r.t. to the i^{th} viewpoint. The set of possible consequences w.r.t. viewpoint i is denoted by X_i and thus vectors x belong to the Cartesian product $\mathcal{X} = X_1 \times \ldots \times X_n$. Also, the notation X_{-i} denotes the Cartesian product $\prod_{j \in N \setminus i} X_j$. Additionally, as in Subsection 1.3 of Chapter 1, the elements of X_i are assumed to be ordered according to a weak order $\succsim_i$ such that for any $x_i, x_i' \in X_i$, $x_i \succsim_i x_i'$ reads "x_i is a better consequence than x_i' w.r.t. viewpoint i". *Marginal utilities* are then defined as real-valued functions $u_i : X_i \to \mathbb{R}, i = 1, \ldots, n$ representing these orders, i.e., such that for any $x_i, x_i' \in X_i$, $x_i \succsim_i x_i' \iff u_i(x_i) \geq u_i(x_i')$. Necessarily, u_i is increasing with $\succ_i$ and defined up to a positive affine translation $\alpha_i u_i + \beta_i$ with $\alpha_i > 0$. Thus, among the infinite possibilities, in what follows we consider functions u_i valued in a common interval $[a, b]$ (potentially obtained from any utility function compatible with $\succsim_i$ with an appropriate choice of α_i, β_i). Note that this setting requires assuming that $\succsim_i$ can be represented by a bounded utility function. Finally, utilities $u_i, i = 1, \ldots, n$ are assumed to be *commensurate* (see Remark 1.4).

Using these notations, the *Choquet integral of marginal utilities* (CIU), denoted by h_w^u, is the following value function:

Definition 2.1 (Choquet integral of utilities (CIU)). *For any $x \in \mathcal{X}$, $h_w^U(x) = C_w(U(x)) = C_w(u_1(x_1), \ldots, u_n(x_n))$, where $U = (u_1, \ldots, u_n)$ is a vector of marginal utilities $u_i, i = 1, \ldots, n$, w is a capacity (see Definition 1.8) and C_w is the Choquet integral w.r.t. w (see Definition 1.9).*

Additionally, in what follows, for any $x \in \mathbb{R}$, x^+ and x^- respectively denotes $x^+ = \max\{0, x\}$ and $x^- = \max\{0, -x\}$. Also, by convention, the notation $S \subseteq N$ excludes the empty set. Moreover, a "non-decreasing" (resp. "increasing") function of one variable is a function that remains constant or increases (resp. increases) when this variable increases. Similarly, real number x is said "non-negative" (resp. "positive") if $x \geq 0$ (resp. $x > 0$).

1 Learning Marginal Utilities

1.1 Decision-Making Under Uncertainty

In *decision-making under uncertainty* (DMU) [Savage, 1954], alternatives are *acts* described by their *outcomes* in the n possible *states of nature*, which represent all the possible scenarios (concerning the object the DM is uncertain about) (see also [Gonzales and Perny, 2020]). In this setting, N is thus the set of all states of nature, and any subset $S \subseteq N$ is an *event*. For instance, in the context of an economic policy choice, the DM may be uncertain about the evolution of an international conflict which is decisive for the policy's outcome. There could then be three states of nature: outbreak (1), status-quo (2) or stabilization of the conflict (3). Then $N = \{1, 2, 3\}$ and for instance, the set $S = \{1, 3\}$ represents the event "$s = 1$ or $s = 3$" (i.e., outbreak or stabilization of the conflict) where s is the actual state of nature. Note that a state of nature corresponds to an *elementary event* in probability theory (i.e., one specific possible result of a random experiment).

The outcomes of the acts in the n states of nature are usually regarded as *payoffs*. For instance, in the example of the choice of an economic policy, the outcomes could be the payoffs generated by the policy expressed in some currency. Let X denote the set of possible payoffs, which for simplicity is assumed to be the real line in the following. In this setting, the set X_i of possible consequences of an alternative w.r.t. to the i^{th} viewpoint (i.e., the i^{th} state of nature) equals the set of possible payoffs X for any $i \in N$ and alternatives are described by payoff vectors $x = (x_1, \dots, x_n) \in \mathcal{X} = X^n$. Therefore, the CIU model involves a unique marginal utility function $u : X \to \mathbb{R}$, describing the DM's attractivity w.r.t. payoffs. As attractivity naturally increases with payoffs, elements of X are ordered by "the higher the better" and u is supposed to be an increasing function.

The role of function u is well understood and the decision behavior of an individual can be interpreted by its analysis. For instance, in *expected utility theory* [Von Neumann and Morgenstern, 1944], which pertains to *decision-making under risk* where outcome probabilities are known, the DM compares *lotteries* according to their expected utility. More formally, let $l = (x_1; p_1, \dots, x_n; p_n)$ denotes the lottery yielding payoff x_i with probability p_i, where outcomes have been ordered (i.e., $x_1 \leq \dots \leq x_n$). Then, the expected utility of l, denoted by $EU(l)$, is defined by $EU(l) = \sum_{i=1}^{n} u(x_i)p_i$. In this setting, *risk aversion* (preference for certain payoffs over uncertain payoffs with known probabilities), is equivalent to the concavity of u and the level of risk aversion of an individual can be measured from the curvature and the slope of his/her utility function [Arrow, 1971, Pratt, 1978].

Choquet expected utility Preference model CIU is known in DMU under the name of *Choquet expected utility* (CEU) [Schmeidler, 1989]. CEU combines the utility function u, describing the DM's sensitivity towards payoffs, with a capacity w, describing the DM's sensitivity towards uncertainty (also known as *chance attitude* [Wakker, 2001]), to assign to any alternative $x \in \mathcal{X}$ the score $h_w^U(x)$ with $U = (u, \dots, u)$, denoted by $h_w^U(x)$.

For any event S, $w(S)$ can be regarded as the likelihood attached to event S by the DM (also known as *subjective probability* [**?**]). When w is *additive*, i.e., $w(S) = \sum_{i \in S} w(\{i\})$ for any $S \subseteq N$, it consists in a standard probability distribution over the set of states of nature. In this case, CEU boils down to EU, i.e., $h_w^U(x) = \sum_{i=1}^{n} w(\{i\})u(x_i)$ where $w(\{i\})$ is the subjective probability of the i^{th} state of nature. In contrast to decision-making under risk, these subjective probabilities are not known and have to be derived from the DM's preferences over acts. Additionnally, a DM may use a *non-additive* capacity when comparing acts, as illustrated in the following standard urn example due to [Ellsberg, 1961].

Example 2.1. *An urn contains 90 balls including 30 red, and 60 blue or yellow balls in unknown proportion. We consider four bets, on the one hand x (resp. y) yielding 100\$ if the drawn ball is red (resp. blue), and on the other hand z (resp. t) yielding 100\$ if the drawn ball is not blue (resp. not red). Here $N = \{R, B, Y\}$ for red, blue, yellow, and the acts under consideration are $x = (100, 0, 0)$, $y = (0, 100, 0)$, $z = (100, 0, 100)$ and $t = (0, 100, 100)$. Note that the pair (x, y) compares similarly* *to the pair (z,t) except that the common outcome attached to yellow balls moves from 0 to 100\$. Despite this similarity, most of people prefer x to y but t to z. Such preferences can not be represented using an additive capacity (i.e., EU) since it would lead to: $w(\{R\})u(100) + w(\{B\})u(0) \geq w(\{R\})u(0) + w(\{R\})u(100)$ on one side and $w(\{R\})u(100) + w(\{B\})u(0) < w(\{R\})u(0) + w(\{R\})u(100)$ on the other.*

However, these preferences can be represented by CEU using a non-additive capacity. Let us assume that $u(0) = 0$ and $u(100) = 1$ and $w(\{R\}) = 1/3$, $w(\{B\}) = w(\{Y\}) = 0$, $w(\{R,B\}) = w(\{R,Y\}) = 1/3$, $w(\{B,Y\}) = 2/3$ and $w(\{R,B,Y\}) = 1$. Note that for all events, w yields the lower possible probability of the event according to our knowledge of the urn content. We have $h_w^U(x) = 0w(\{R,B,Y\}) + (1-0)w(\{R\}) = 1/3$. Similarly we obtain $h_w^U(y) = 0$, $h_w^U(z) = 1/3$ and $h_w^U(t) = 2/3$. Hence, $h_w^U(x) > h_w^U(y)$ and $h_w^U(t) > h_w^U(z)$ which is consistent with the observed preferences.

Remark 2.1 (interactions in DMU). It could have also been observed that preferences $x \succsim y$ and $z \prec t$ in Example 2.1 consist in an example of mutual preferential independence violation for $S = \{R, B\}$ (see Remark 1.2), thus indicating that no additive value function can represent them. As illustrated in multi-criteria/attribute decision-making in Subsection 1.3 of Chapter 1, the potential non-additivity of capacity w allows CEU to bypass this descriptive limitation by accounting for interactions between states of nature in the individual's uncertainty perception. For instance, in Example 2.1, the probability of the events $\{R\}$ and $\{B\}$ are imprecisely known (between 0 and $\frac{2}{3}$) while the probability of the event $\{R,B\}$ is known and equal to $\frac{2}{3}$.

The interesting properties of the Choquet integral, already discussed in Section 1.2 of Chapter 1, have been extensively studied in the DMU literature to understand the CEU model. For instance, CEU satisfies *monotonicity* (see Definition 1.6), i.e., for any pair of act $x, x' \in \mathcal{X}$, if $x_i \geq x'_i$ for any $i \in N$ then $h_w^U(x) \geq h_w^U(x')$. Also, preferences induced by CEU satisfy *uncertainty aversion* if and only if u is concave and w is supermodular (see Definition 1.13) [Chateauneuf and Tallon, 2002a]. Uncertainty aversion (also known as *convexity of preferences*) is the tendency to prefer certain payoffs over uncertain payoffs. More formally, for any $\alpha \in [0,1]$, if the DM is indifferent between x and x' then $\alpha x + (1-\alpha)x'$ will be preferred to x (and also to x' by symmetry). The convex mixture of x and x' reduces the uncertainty of outcomes w.r.t x and x', making the DM better off. Note that this property follows from the fact that the Choquet integral associated with a supermodular capacity favors balanced solutions [Lesca and Perny, 2010], as illustrated in Example 1.5 of Chapter 1.

Additionally, in decision-making under risk, CEU boils down to the *rank-dependent utility* (RDU) model whenever, for any $S \subseteq N$, $w(S) = g(\sum_{i \in S} p_i)$ where $p_1, \ldots, p_n$ are the outcomes' probabilities and g is a monotonic weighting function such that $g(0) = 0$ and $g(1) = 1$[Quiggin, 2012]. It is formally defined as follows:

Definition 2.2 (rank-dependent utility (RDU)). *For any lottery* $l = (x_1; p_1, \dots, x_n; p_n)$ *such that* $x_1 \leq \dots \leq x_n$*,*

$$RDU(l) = \sum_{i=1}^{n}(g(\sum_{k=i}^{n} p_k) - g(\sum_{k=i+1}^{n} p_k))u(x_i)$$

where $g : [0, 1] \rightarrow [0, 1]$ *is a non-decreasing function such that* $g(0) = 0$ *and* $g(1) = 1$*.*

If in addition, u is linear then CEU boils down to Yaari's model [Yaari, 1987]. Below, we present standard methods in decision-making under risk for eliciting the utility function in the EU and RDU model, respectively known as the *certainty-equivalent method* [Von Winterfeldt and Edwards, 1986] and *tradeoff method* [Wakker and Deneffe, 1996, Abdellaoui, 2000]. The latter method is then exploited for eliciting the utility function in the CEU model.

Certainty-equivalent method within EU theory A standard method to elicit the utility function in the EU model is to rely on *certainty-equivalent* queries involving lotteries. Such queries take the following form: "for two outcomes $o_1, o_2 \in X$ such that $o_1 \leq o_2$ and a probability $p \in [0, 1]$, what is the outcome o_3 such that $(o_3; 1) \sim (o_1; p, o_2; 1 - p)$?". Outcome o_3 is then said to be the *certainty equivalent* of lottery $l = (o_1; p, o_2; 1 - p)$ and the indifference statements yields $EU((o_3; 1)) = EU(l)$ which is equivalent to $u(o_3) = pu(o_1) + (1 - p)u(o_2)$. As u is defined up to a positive affine transformation, one can start with two outcomes $o_1, o_2 \in X$ whose values are arbitrarily set to $u(o_1) = 0, u(o_2) = 1$ and a probability $p = \frac{1}{2}$, and obtain a first point $u(o_3) = \frac{1}{2}$. The utility curve is then usually incrementally constructed using a dichotomic scheme that asks the certain-equivalent of $(o_1; \frac{1}{2}, o_3; \frac{1}{2})$ on one side and of $(o_3; \frac{1}{2}, o_2; \frac{1}{2})$ on the other, and so on [Von Winterfeldt and Edwards, 1986].

Tradeoff method in the RDU model If we now consider the RDU model with a weighting function g, the knowledge of the certainty equivalent o_3 of the two-outcome lottery $(o_1; p, o_2; 1 - p)$ gives us $u(o_3) = g(p)u(o_1) + (1 - g(p))u(o_2)$. However, since g is unknown, this equality no longer allows us to construct the utility curve as in the certainty-equivalent method. A way to disentangle parameter u and g is to exploit the *tradeoff* method [Wakker and Deneffe, 1996, Abdellaoui, 2000, Hines and Larson, 2010, Perny et al., 2016]. This method relies on tradeoff queries of the following form: "for an outcome $o_1 \in X$, two reference outcomes $r, R \in X$ such that $r < R$ and a probability

$p \in [0,1]$, what is the outcome o_2 such that $(1-p; o_1, p; R) \sim (1-p; o_2, p; r)$?"

Such query, denoted by $Q(o_1)$, asks the DM how much of an increase in the outcome with probability $1-p$ would be needed to compensate for a loss of the outcome with probability p. Using this indifference statement, we obtain that $RDU((1-p; o_1, p; R)) = RDU((1-p; o_2, p; r))$ which is equivalent to $(u(o_1) - u(o_2))(1-g(p)) = g(p)(u(r) - u(R))$. Finally, using a second query $Q(o_2)$ whose answer is o_3, if $0 < g(p) < 1$, we obtain the following equality: $u(o_1) - u(o_2) = u(o_2) - u(o_3)$.

Then, these queries are used to elicit u within a given interval $[\mathbf{0}, M] \subseteq X$ (where $\mathbf{0}, M$ are two outcomes such that $\mathbf{0} < M$) by taking $o_1 = \mathbf{0}$ and r, R such that $\mathbf{0} < M < r < R$, and incrementally construct a standard sequence of outcomes $\{o_1, o_2, ..., o_q\}$ where o_t is the answer to $Q(o_{t-1})$. The sequence stops at step $t = q$ when $o_t \geq M$ and by construction, $u(o_{t+1}) - u(o_t) = u(o_t) - u(o_{t-1})$, $t = 2, \ldots, q-1$. Again, one can arbitrarily set $u(o_1) = 0$ and $u(o_q) = 1$ and the utility function is thus completely determined on the points of the standard sequence, i.e., $u(o_t) = (t-1)/(q-1)$, $t = 1, \ldots, q$.

In the following, *we exploit the tradeoff method principle to derive a method for learning the utility function in the CEU model.* First, we propose tradeoff queries adapted to the CEU model, and then we formulate a monotonic regression problem to learn the utility function from the obtained tradeoff indifference statements.

1.1.1 Tradeoff Queries in CEU Theory

Tradeoff queries The counterpart in DMU (where outcomes' probabilities are unknown) of the two-outcome lotteries used in decision-making under risk are the *mixtures of constant acts.* For any event $S \subseteq N$ and any acts $x, x' \in \mathcal{X}$, the *mixture of acts* xSx' whose outcome in the i^{th} state of nature is x_i if $i \in S$ and x'_i otherwise. Moreover, a *constant act* is an act whose outcome does not depend on the state of nature. For any outcome $o \in X$, the constant act of outcome o is denoted by $\bar{o} = (o, \ldots, o)$. Then, for any $o_1, o_2 \in X$ and any $S \subseteq N$, the mixture of constant acts $\bar{o_1} S \bar{o_2}$ is the act yielding outcome o_1 when S occurs and o_2 otherwise.

For any outcome $o \in X$, it can easily be checked that the overall value of the constant act $\bar{o}$ under CEU equals o, i.e., $h_w^U(\bar{o}) = o$, provided $w(N) = 1$. Then, asking the certainty equivalent $o_3 \in X$ of the act $\bar{o_1} S \bar{o_2}$ for $o_1, o_2 \in X$, yields the indifference $\bar{o_1} S \bar{o_2} \sim \bar{o_3}$ which is equivalent to $h_w^U(\bar{o_1} S \bar{o_2}) = w(S)u(o_1) + (1 - w(S))u(o_2) = u(o_3)$. However, similarly to the case of the RDU model, such equality does not allow to elicit the utility function since capacity w is unknown. For this reason, we propose to apply the principle of the tradeoff method to isolate u within CEU.

As in the tradeoff method, u is elicited within a given interval $[\mathbf{0}, M] \subseteq X$ where $\mathbf{0}, M \in X$ are two outcomes such that $\mathbf{0} < M$. The proposed method requires the existence of an event S such that $\bar{\mathbf{0}} \prec \bar{\mathbf{0}}S\bar{M} \prec \bar{M}$. Within CEU theory these strict preferences translate into $h^U_w(\bar{\mathbf{0}}) < h^U_w(\bar{\mathbf{0}}S\bar{M}) < h^U_w(\bar{M})$ which is equivalent to $u(\mathbf{0}) = h^U_w(\bar{\mathbf{0}}) < u(\mathbf{0})(1 - w(\bar{S})) + u(M)w(\bar{S}) < h^U_w(\bar{M}) = u(M)$, i.e., $0 < w(\bar{S}) < 1$ since $u(\mathbf{0}) < u(M)$. The following property shows that, under this assumption, tradeoff queries formulated with mixtures of constant acts can be used to derive constraints on the utility function within CEU theory:

Proposition 2.1. *Let $S \subseteq N$ such that $\bar{\mathbf{0}} \prec \bar{\mathbf{0}}S\bar{M} \prec \bar{M}$, and consider outcomes $o_1, r, R \in X$ such that $\mathbf{0} \leq o_1 < M < r < R$. If the two following queries are successively asked:*

- *$Q_S(o_1|r, R)$: "what is the outcome o_2 such that: $\bar{o_1}S\bar{r} \sim \bar{o_2}S\bar{R}$?"*

- *$Q_S(o_2|r, R)$: "what is the outcome o_3 such that: $\bar{o_2}S\bar{r} \sim \bar{o_3}S\bar{R}$?"*

then, the following equality holds:

$$u(o_1) - u(o_2) = u(o_2) - u(o_3) \tag{2.1}$$

Proof. As $r < R$, we necessarily have $o_2 \leq o_1$ and $o_3 \leq o_2$ (otherwise we would have $\bar{o_1}S\bar{r} \prec \bar{o_2}S\bar{R}$ and $\bar{o_2}S\bar{r} \prec \bar{o_3}S\bar{R}$). Therefore $o_3 \leq o_2 \leq R$ holds. Thus we have $h^U_w(\bar{o_2}S\bar{R}) = u(o_2)(1 - w(\bar{S})) + u(R)w(\bar{S})$ and $h^U_w(\bar{o_1}S\bar{r}) = u(o_1)(1 - w(\bar{S})) + u(r)w(\bar{S})$. Hence, since the first indifference statement yields $h^U_w(\bar{o_2}S\bar{R}) = h^U_w(\bar{o_1}S\bar{r})$, we have $u(o_2)(1 - w(\bar{S})) + u(R)w(\bar{S}) = u(o_1)(1 - w(\bar{S})) + u(r)w(\bar{S})$ and therefore $(1 - w(\bar{S}))[u(o_1) - u(o_2)] = w(\bar{S})[u(R) - u(r)]$. Similarly, the second indifference statements implies $(1 - w(\bar{S}))[u(o_2) - u(o_3)] = w(\bar{S})(u(R) - u(r)]$. Finally, we have $(1 - w(\bar{S}))[u(o_1) - u(o_2)] = (1 - w(\bar{S}))[u(o_2) - u(o_3)]$ and since $w(\bar{S}) < 1$, we obtain Equation 2.1.

As illustrated in Figure 2.1, with such indifferences, the DM makes a tradeoff between downgrading o_1 in o_2 (or o_2 in o_3) if event S occurs and upgrading r in R if event S does not occur. These tradeoff queries are referred to as *Q-queries* in the following.

Short standard sequences Similarly to the tradeoff method, such queries could be used to construct a standard sequence by sequentially asking queries $Q_S(o_t|R, r)$. More precisely, one could start with $o_1 = \mathbf{0}$, take o_{t+1} as the answer to query $Q_S(o_t|R, r)$, and stops at step $t = q$ when $o_t \geq M$. By construction, this sequence is such that for any $t = 1, \dots, q$, $o_t \leq o_{t+1}$ and $u(o_{t+1}) - u(o_t) = u(o_t) - u(o_{t-1})$ by Equation 2.1. By letting $u(o_1) = 0$ and $u(o_q) = 1$, the utility function is completely determined on the standard sequence, as in the tradeoff method for RDU, i.e., $u(o_t) = (t-1)/(q-1)$ for $t = 1, \dots, q$.

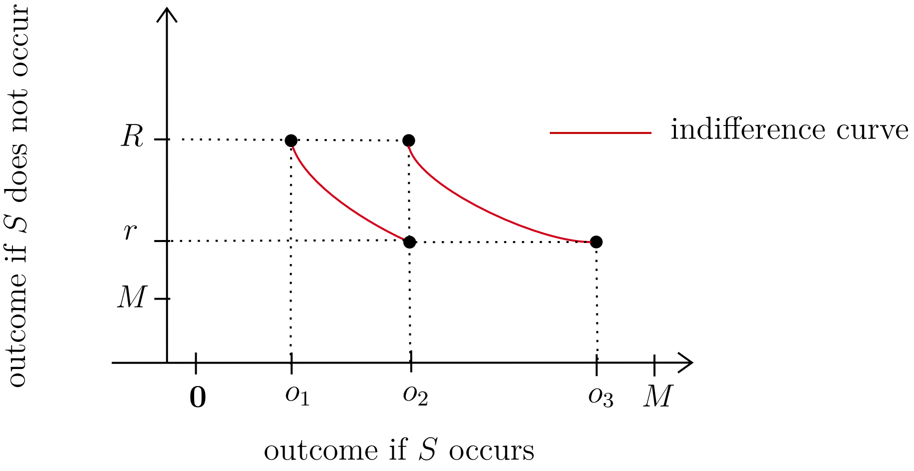


Figure 2.1: Indifferences obtain with $Q_S(o_1|r, R)$ and $Q_S(o_2|r, R)$ queries.

However, if the DM makes some errors in assessing o_t in the early steps of the sequence, these errors will propagate and impact the whole sequence [Blavatskyy, 2006]. Since the noise distortion naturally increases error with the length of the standard sequence, we propose an alternative approach that relies on multiple minimal length ($q = 2$) standard sequences of type (o_1, o_2, o_3). Multiplicity is obtained by varying the initial location o_1, and the mesh (r, R). Putting all together, we obtain a database $D = (o_1^\ell, o_2^\ell, o_3^\ell)_{\ell=1}^T$ associated with the linear constraints:

$$u(o_1^\ell) - u(o_2^\ell) = u(o_2^\ell) - u(o_3^\ell), \ell = 1, \ldots, T \tag{2.2}$$

Then, we propose to perform a *spline* regression to identify the utility function that best fits the set of linear constraints. In particular, given that the utility function is non-decreasing, we suggest employing a basis of *I-spline* functions, which are smooth, non-negative, and monotonic (non-decreasing).

1.1.2 Monotonic Spline Regression

Spline functions A *spline* function of order k, $k \in \mathbb{N}^*$ is a function that is piecewise polynomial of degree less than or equal to k, and of class C^{k-1} (i.e., whose derivatives up to order $k - 1$ are continuous) [De Boor, 1978]. Spline functions are widely used for data interpolation or approximation due to their ability to smoothly approximate complex shapes (see for instance [Hastie et al., 2009]-Chapter 5). Moreover, they allow for a compact representation of value functions. Indeed, a spline function can be expressed as a linear combination of basis functions and is thus characterized by the coefficients of the combination. Since the utility function increases with payoffs, we will use a basis of non-decreasing spline functions, known as *I-spline* functions [Ramsay, 1988]. I-spline

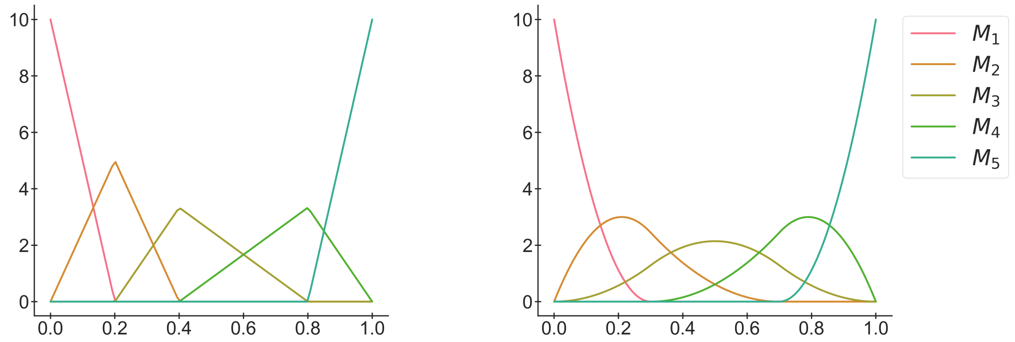


Figure 2.2: Examples of M-basis of size $m = 5$ for $k = 2$ (left) and $k = 3$ (right).

functions are built upon *M-spline* functions, which we introduce below.

A M-spline function of order k, $k \in \mathbb{N}^*$, is a non-negative and piecewise polynomial function where each piece is of degree less than or equal to $k-1$. A m-dimensional basis of k-order M-Spline functions defined over an interval $[a,b]$ can be constructed using a set of knots $t = \{t_1, \dots, t_{m+k}\}$ such that:

- $t_1 = \dots = t_k = a$
- $t_{m+1} = \dots = t_{m+k} = b$
- $t_l < t_{l+k},\ \ l = 1, \dots, m$

Then, the basis functions $M_l, l = 1, \dots, m$ are defined for any $x \in [a,b]$ as follows:

$$M_l(x \mid k, t) = \begin{cases} \frac{1}{t_{l+1}-t_l} & \text{if } k = 1 \text{ and } t_l \leq x < t_{l+1} \\ \frac{(k\cdot(x-t_l)\cdot M_l(x|k-1,t)+(t_{l+k}-x)\cdot M_{l+1}(x|k-1,t))}{(k-1)\cdot(t_{l+k}-t_l)} & \text{if } k > 1 \\ 0 & \text{otherwise.} \end{cases}$$

Functions $M_1, \dots, M_m$ form a basis for the vector space of functions defined on $[a,b]$, polynomial of degree less than or equal to $k-1$ on each interval $[t_l, t_{l+1}[$ and of class C^{k-1-r} in the neighborhood of each knot of multiplicity r (number of times the knot appears in the subdivision). In the following, we only consider the case $r = 1$ for the inner knots of the subdivision (i.e., $t_{k+1}, \dots, t_m$), and thus functions M_l are of class C^{k-2} within $]a,b[$. As an illustration, the M-basis for $m = 5$, $k = 2$ and $t = (0, 0, 0.2, 0.4, 0.8, 1, 1)$ is represented on the left side of Figure 2.2 and for $m = 5$, $k = 3$ and $t = (0, 0, 0, 0.3, 0.7, 1, 1, 1)$ on the right side.

Then, m-dimensional basis of k-order I-Spline functions defined on the interval $]a,b[$ can be constructed using a set of knots $t = \{t_1, \dots, t_{m+k+2}\}$ such that:

- $t_1 = \dots = t_{k+1} = a$

- $t_{m+2} = \cdots = t_{m+k+2} = b$
- $t_l < t_{l+k},\ l = 1, \ldots, m+1$

The I-spline basis functions, denoted by $I_l, l = 1, \ldots, m$, are then defined for any $x \in [a, b]$ as follows:

$$I_l(x \mid k, t) = \int_a^x M_l(u \mid k, t)\mathrm{d}u = \begin{cases} 0 & \text{if } j < i, \\ 1 & \text{if } j > i+k-1, \\ \sum_{z=i}^{j} \frac{t_{z+k+1}-t_z}{k+1} M_z(x \mid k+1, t) & \text{otherwise.} \end{cases}$$

where j is the index of t such that $t_j \leq x < t_{j+1}$. Note that by convention, function I_l is defined on the boundaries of the interval by $I_l(a) = 0$ and $I_l(b) = 1$.

As integrals of M-spline functions that are non-negative functions of class C^{k-2}, I-spline functions are non-negative and non-decreasing functions of class C^{k-1}. Then, smooth and non-decreasing functions v_α can be generated using linear combinations of the I-spline basis with non-negative coefficients $\alpha = (\alpha_1, \ldots, \alpha_m) \in \mathbb{R}_+^m$ as follows:

$$v_\alpha(x) = \sum_{l=1}^{m} \alpha_l I_l(x), \text{ for any } x \in [a, b]\,. \tag{2.3}$$

For the sake of illustration, the I-spline basis for $m = 5$, $k = 3$ and $t = (0, 0, 0, 0.2, 0.5, 0.8, 1, 1, 1)$ is represented in Figure 2.3 along with an instance of function v_α for $\alpha = (0.2, \ldots, 0.2)$.

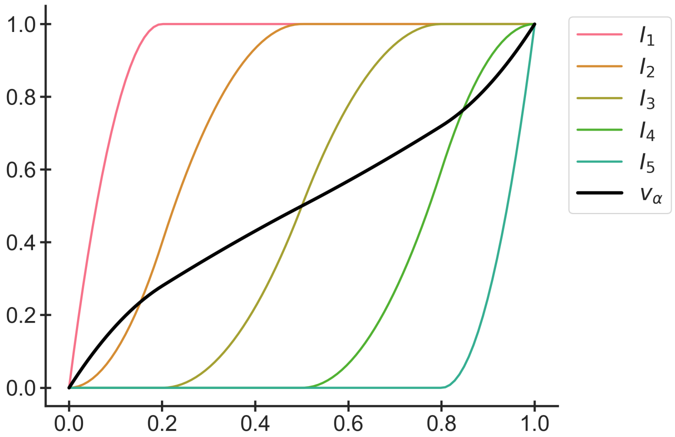


Figure 2.3: Example of I-spline basis of size $m = 5$ for $k = 3$ and v_α for $\alpha = (0.2, \ldots, 0.2)$

Regression problem Then, we propose to perform a regression for learning the utility function using the parametric model given by Equation 2.3 for I-spline functions defined over $[a, b] = [\mathbf{0}, M]$. More precisely, we want to determine the parameter α that best fits the set of constraints associated with the database $D = (o_1^\ell, o_2^\ell, o_3^\ell)_{\ell=1}^T$ given by Equation 2.2. The regression problem is thus formulated as follows:

$$\min_{\alpha \in \Delta_m} \sum_{\ell=1}^{T} |2v_\alpha(o_2^\ell) - v_\alpha(o_1^\ell) - v_\alpha(o_3^\ell))| \tag{2.4}$$

Here Δ_m denotes the simplex of size m, i.e., $\{\alpha \in \mathbb{R}_+^m | \sum_{l=1}^m \alpha_l = 1\}$, where the sum constraint guarantees that $v_\alpha(M) = 1$ since $I_l(M) = 1, l = 1, \dots, m$. Note that $v_\alpha(\mathbf{0}) = 0$ since $I_l(\mathbf{0}) = 0, l = 1, \dots, m$.

Remark 2.2 (linearization of the absolute value). Throughout the thesis, we use a standard trick to reformulate minimization problems involving absolute values in the objective as linear programs. This trick relies on the following remark: for any $x \in \mathbb{R}$, there exists $x^+, x^- \in \mathbb{R}_+$ such that $x = x^+ - x^-$. Among the possible pairs (x^+, x^-), the one that minimizes the sum $x^+ + x^-$ is such that $x^- = 0$ if $x \geq 0$ and $x^+ = 0$ if $x \leq 0$, which is equivalent to $x^+ + x^- = |x|$. Then minimizing $|x|$ boils down to solving the following linear program:

$$\begin{aligned} \min_{x, x^+, x^-} \; & x^+ + x^- \\ \text{s.t. } & x^+ - x^- = x \end{aligned}$$

Hence, using linearization variables $\epsilon_\ell^+, \epsilon_\ell^-$, to model the constraint violation $|2v_\alpha(o_1^\ell) - v_\alpha(o_2^\ell) - v_\alpha(o_0^\ell))|$, the problem can be formalized as the following linear program with $T+1$ constraints and $m + 2T$ variables:

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^m} z = & \sum_{\ell=1}^{T} (\epsilon_\ell^+ + \epsilon_\ell^-) \\ & \sum_{l=1}^{m} \alpha_l (2I_l(o_2^\ell) - I_l(o_1^\ell) - I_l(o_3^\ell)) = \epsilon_\ell^+ - \epsilon_\ell^-, \quad \ell = 1, \dots T \\ & \sum_{l=1}^{m} \alpha_l = 1 \\ & \epsilon_\ell^+ \geq 0, \epsilon_\ell^- \geq 0, \ell = 1, \dots, T \\ & \alpha_l \geq 0, l = 1, \dots, m \end{aligned} \tag{2.5}$$

Hereafter let α^* denote the optimal solution, z^* the optimal value of Problem 2.5.

Uncertainty quantification on the learned utility functions Taking into consideration that the number of observations may be limited (as the DM may not be able to answer a high number of Q-queries), we need to assess the level of uncertainty on the learned marginal utility function. To this end, we examine a neighborhood of the optimal solution defined by $z^* \leq z \leq z^* + \delta$ where δ is a tolerance threshold, and denoted by $V_\delta(z^*)$. This neighborhood contains all parameters α such that v_α satisfies the constraints associated with database D with an error z at most equal to $z^* + \delta$.

The range of variation of v_α within $V_\delta(z^*)$ is a good indicator of the level of uncertainty allowed by the constraints. It can be measured by the following quantity:

$$\rho = \max_{o \in [\mathbf{0}, M]} \{ \max_{\alpha \in V_\delta(z^*)} v_\alpha(o) - \min_{\alpha \in V_\delta(z^*)} v_\alpha(o) \}$$

which may be estimated by discretization of $[\mathbf{0}, M]$. When ρ is too large (higher than a predetermined threshold denoted by ϵ), the constraints are considered too weak to allow for the identifiability of the utility function; one should carry on the Q-queries process. The overall proposed learning procedure is summarized in Algorithm 2.1.

Algorithm 2.1: utility function learning with Q-queries

```
Inputs: $\mathbf{0}, M, \epsilon, \delta$
$\ell \leftarrow 1$, $D \leftarrow \emptyset$
while $\rho \leq \epsilon$ do
    Select $S^\ell$ such that $\bar{\mathbf{0}} \prec \bar{\mathbf{0}} S^\ell \bar{M} \prec \bar{M}$
    Select $o_1^\ell, R^\ell, r^\ell$ such that $\mathbf{0} \leq o_1^\ell < M < R^\ell < r^\ell$
    $o_2^\ell \leftarrow$ answer to query $Q_{S^\ell}(o_1^\ell | R^\ell, r^\ell)$
    $o_3^\ell \leftarrow$ answer to query $Q_{S^\ell}(o_2^\ell | R^\ell, r^\ell)$
    $D \leftarrow D \cup \{(o_1^\ell, o_2^\ell, o_3^\ell)\}$
    $(\alpha^*, z^*) \leftarrow$ solution and optimal value of Problem 2.5 with database $D$
    $\rho \leftarrow \max_{o \in [\mathbf{0}, M]} \{\max_{\alpha \in V_\delta(z^*)} v_\alpha(o) - \min_{\alpha \in V_\delta(z^*)} v_\alpha(o)\}$
    $\ell \leftarrow \ell + 1$
Outputs: $\alpha^*$
```

The proposed procedure enables the learning of the utility function on a portion $[\mathbf{0}, M]$ of the set of possible payoffs X using reference payoffs r, R higher than M. In the case where the set of possible payoffs is a bounded interval $[a, b]$ on which we want the utility function to be completely determined, one can first use the proposed procedure for $\mathbf{0} = a, M = \frac{a+b}{2}$ using reference outcomes r, R within $[\frac{a+b}{2}, b]$ and then use a symmetrical learning procedure to learn the marginal utility on $[a, \frac{a+b}{2}]$.

Remark 2.3 (Rashomon set). The set of model parameters $V_\delta(z^*)$ can be regarded as a *Rashomon set*, formally defined as the set of all models that exhibit near-optimal accuracy [Breiman, 2001]. This set is named after the *Rashomon effect* which describes how different people can have contradictory interpretations of the same event. The Rashomon set is a particularly relevant notion in interpretability in machine learning as a rich Rashomon set could lead to various model interpretations [Semenova et al., 2022].

1.2 Multi-criteria/attribute Decision-Making

In this section, the problem of learning marginal utilities within the CIU model is addressed in the general context of *multiattribute decision-making* [Keeney and Raiffa, 1976, Dyer, 2005] (MADM), including in particular *multicriteria decision-making* (MCDM) [Roy and Vincke, 1981, Grabisch, 2016b]. In MADM, alternatives are described by n *attributes*, associated with no particular semantics. In MCDM, alternatives are described by their *performances* w.r.t. n *criteria*, which are n different ways of evaluating the alternatives. More formally, a criterion can be defined as a real-valued function that assigns to each alternative a measure of its performance w.r.t. a certain viewpoint. For instance, an economic policy could be evaluated according to three criteria: economic efficiency, social impact, and environmental impact.

In the MADM/MCDM setting, the set of alternative $\mathcal{X}$ thus takes the form of an heterogenous Cartesian product $\mathcal{X} = X_1 \times \ldots \times X_n$. There is therefore n marginal utilities $u_i : X_i \to \mathbb{R}, i = 1, \ldots, n$ to be elicited. For the sake of generality, we propose a method for learning marginal utilities within the *bipolar Choquet integral of marginal utilities* (bi-CIU), which is a generalization of the CIU model allowing for modeling distinct behavior in the face of "good" or "bad" consequences. Using this model thus requires obtaining in close cooperation with the DM, for any viewpoints $i \in N$, a *neutral element*, denoted by $\mathbf{0}_i$, separating "good" and "bad" consequences within the consequence set X_i. Also, we distinguish within X_i two elements referred to as the *bottom level* and the *top level* consequences, respectively denoted by $-\mathbf{1}_i$ and $\mathbf{1}_i$. For any $i \in N$, marginal utility u_i is increasing with $\succ_i$ and consequences above the neutral level receive a positive marginal value whereas consequences below the neutral level receive a negative marginal utility, i.e., $u_i(-\mathbf{1}_i) = -1$, $u_i(\mathbf{0}_i) = 0$ and $u_i(\mathbf{1}_i) = 1$.

In this setting, the bi-CIU model, denoted by $h^U_{w,w'}$, is formally defined as follows:

Definition 2.3 (bi-CIU model). *For any $x \in \mathcal{X}$, $h^U_{w,w'}(x) = BC_{w,w'}(u(x))$ where $U = (u_1, \ldots, u_n)$ is a vector containing marginal utilities $u_i, i = 1, \ldots, n$ are n, (w, w') are two capacities and $BC_{w,w'}$ is the bipolar-Choquet integral w.r.t. (w, w') (see Definition 1.14).*

The proposed method consists in learning independently the n marginal utilities $u_i, i = 1, \ldots, n$ using for each u_i, a method inspired from the learning of the utility function in the CEU model. Similarly to the utility function learning, the proposed method first derives constraints on u_i using a specifically designed elicitation process and then performs a monotonic spline regression. *Also relying on tradeoff queries, the elicitation process differs from the one used in DMU in that it not only allows for separating the effect of marginal utilities from that of capacities in the Choquet integral but also isolates*

u_i from the other marginal utilities u_j, $j \neq i$.

1.2.1 Tradeoff Queries in the bi-CIU Model

Let i be any element of N. The proposed elicitation process to derive constraints on u_i involves tradeoffs between attribute i and another attribute j of N that can be freely chosen. Starting from an alternative $x \in \mathcal{X}$ and considering a given modification of the j^{th} attribute value (i.e., of component x_j), the tradeoff query consists in asking which variation of the i^{th} attribute value (i.e., of component x_i) would exactly compensate the variation of x_j. The existence of answers exactly achieving the compensation requires a certain richness of attribute domain X_i. This assumption is formalized by the *restricted solvability* axiom well known in mathematical psychology [Krantz and Tversky, 1971]. For any two vectors x, x' in $\mathcal{X}$, let (x_i, x'_{-i}) denote the vector derived from x' by substituting the i^{th} component by x_i. Then, restricted solvability can be stated as follows:

Definition 2.4 (restricted solvability). *A preference relation $\succsim$ on $\mathcal{X}$ satisfies restricted solvability w.r.t. the i^{th} component if for any $x \in \mathcal{X}$, $a_i, b_i \in X_i$, $t_{-i} \in X_{-i}$ with $(a_i, t_{-i}) \succsim x \succsim (b_i, t_{-i})$, there exists x'_i such that $x \sim (x'_i, t_{-i})$. When this holds for all $i \in N$, the binary relation is said to satisfy restricted solvability.*

Restricted solvability is not always satisfied, especially in the case of discrete attributes, as shown in the following example.

Example 2.2. *Let $X_1 = \{0,1\}$ and $X_2 = \{0, \frac{1}{2}, 1\}$ and define $\succsim$ on $X_1 \times X_2$ by $(x_1, x_2) \succsim (x'_1, x'_2)$ iff $x_1 + x_2 \geq x'_1 + x'_2$. We have $(1,0) \succsim (0, \frac{1}{2}) \succsim (0,0)$ but there is no $x_1 \in X_1$ such that $(x_1, 0) \sim (0, \frac{1}{2})$. Here restricted solvability does not hold w.r.t. the first component.*

In the following, restricted solvability is assumed to hold. For the sake of readability, the case where this assumption does not hold is detailed in Appendix A.1. Also, the method requires that $(\mathbf{1}_i, \mathbf{0}_{-i}) \succ \mathbf{0} \succ (-\mathbf{1}_i, \mathbf{0}_{-i})$, i.e., $w(\{i\}) > 0 > w'(N \setminus \{i\}) - 1$ under the bi-CIU model. Let us now present the elicitation process to derive constraints on u_i successively below, and above the neutral level $\mathbf{0}_i$.

(i) Marginal utility elicitation below the neutral level The proposed tradeoff queries involve alternatives of the form $(a_i, b_j, \mathbf{0}_{-ij})$, denoting a vector of neutral consequences everywhere except on components i and j where consequences are $a_i \in X_i$ and $b_j \in X_j$.

Proposition 2.2. *For any attribute $j \in N$, let r_j, $R_j \in X_j$, $x_i \in X_i$ such that $\mathbf{0}_j \precsim_j r_j \prec_j R_j$, $x_i \precsim_i \mathbf{0}_i$. If the two following queries are successively asked:*

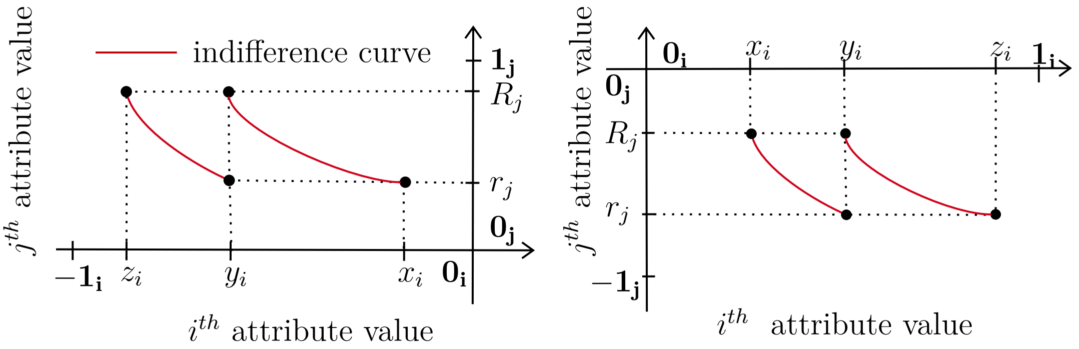


Figure 2.4: Indifferences obtained with Q-queries below (left) and above (right) $\mathbf{0}_i$.

- $Q_{ij}(x_i|R_j,r_j)$: *what is the consequence* y_i *such that* $(x_i,r_j,\mathbf{0}_{-ij}) \sim (y_i,R_j,\mathbf{0}_{-ij})$*?*

- $Q_{ij}(y_i|R_j,r_j)$: *what is the consequence* z_i *such that* $(y_i,r_j,\mathbf{0}_{-ij}) \sim (z_i,R_j,\mathbf{0}_{-ij})$*?*

then, the following equality holds:

$$u_i(x_i) - u_i(y_i) = u_i(y_i) - u_i(z_i) \tag{2.6}$$

Proof. *As* $r_j \prec_j R_j$, *we necessarily have* $y_i \precsim_i x_i$ *and* $z_i \precsim_i y_i$ *otherwise we would have* $(x_i,r_j,\mathbf{0}_{-ij}) \succ (y_i,R_j,\mathbf{0}_{-ij})$ *and* $(y_i,r_j,\mathbf{0}_{-ij}) \succ (z_i,R_j,\mathbf{0}_{-ij})$. *Therefore, since* $x_i \precsim_i \mathbf{0}_i$, *we have* $z_i \precsim_i y_i \precsim_i x_i \precsim_i \mathbf{0}_i$ *and finally since* $\mathbf{0}_j \precsim_j r_j \prec_j R_j$, *we obtain* $u_i(z_i) \leq u_i(y_i) \leq u_i(x_i) \leq 0 \leq u_j(r_j) < u_j(R_j)$. *Thus,* $h^U_{w,w'}(x_i,r_j,\mathbf{0}_{-ij}) = u_j(r_j)w(\{j\}) + u_i(x_i)(1-w'(N\setminus\{i\}))$ *and similarly* $h^U_{w,w'}(y_i,R_j,\mathbf{0}_{-ij}) = u_j(R_j)w(\{j\}) + u_i(y_i)(1-w'(N\setminus\{i\}))$. *Thus, from the first indifference statement,* $(x_i,r_j,\mathbf{0}_{-ij}) \sim (y_i,R_j,\mathbf{0}_{-ij})$, *we have:* $h^U_{w,w'}(x_i,r_j,\mathbf{0}_{-ij}) = h^U_{w,w'}(y_i,R_j,\mathbf{0}_{-ij})$ *and therefore* $(u_i(x_i)-u_i(y_i))(1-w'(N\setminus\{i\})) = (u_j(R_j)-u_j(r_j))w(\{j\})$. *Moreover, using the second indifference* $(y_i,r_j,\mathbf{0}_{-ij}) \sim (z_i,R_j,\mathbf{0}_{-ij})$, *we obtain* $(u_i(y_i) - u_i(z_i))(1-w'(N\setminus\{i\})) = (u_j(R_j)-u_j(r_j))w(\{j\})$. *Then* $(u_i(x_i)-u_i(y_i))(1-w'(N\setminus\{i\})) = (u_i(y_i) - u_i(z_i))(1 - w'(N \setminus \{i\}))$. *Finally, since* $(-\mathbf{1}_i,\mathbf{0}_{-i}) \prec \mathbf{0}$, *i.e.,* $w'(N \setminus \{i\}) < 1$ *we obtain Equation 2.6.*

If we additionally assume that $(\mathbf{0}_i,R_j,\mathbf{0}_{-ij}) \succsim (x_i,r_j,\mathbf{0}_{-ij}) \succsim (-\mathbf{1}_i,R_j,\mathbf{0}_{-ij})$, by considering an instance of the restricted solvability axiom (Definition 2.4) obtained for $a_i = \mathbf{0}_i$, $b_i = -\mathbf{1}_i$, $t_{-i} = (R_j,\mathbf{0}_{-ij})$ and $x = (x_i,r_j,\mathbf{0}_{-ij})$, one can see that an answer $y_i \in X_i$ to question $Q_{ij}(x_i)$ is guaranteed to exist by the restricted solvability assumption. A similar reasoning guarantees the existence of an answer to the second query.

As illustrated in Figure 2.4(left), with such indifferences, the DM makes a tradeoff between upgrading r_j in R_j on the j^{th} attribute and downgrading x_i in y_i (or y_i in z_i) on the i^{th} attribute.

(ii) Marginal utility elicitation above the neutral level The process is symmetric to the one used to elicit u_i below the neutral level.

Proposition 2.3. *For any attribute $j \in N$, let $r_j, R_j \in X_j$ and $x_i \in X_i$ such that $r_j \prec_j R_j \precsim_j \mathbf{0}_j$, $x_i \succsim_i \mathbf{0}_i$. If y_i is the answer to the query $Q_{ij}(x_i|r_j, R_j)$ (i.e., $(x_i, R_j, \mathbf{0}_{-ij}) \sim (y_i, r_j, \mathbf{0}_{-ij})$) and z_i is the answer to the query $Q_{ij}(y_i|r_j, R_j)$ (i.e., $(y_i, R_j, \mathbf{0}_{-ij}) \sim (z_i, r_j, \mathbf{0}_{-ij})$), then, the following equality holds:*

$$u_i(y_i) - u_i(x_i) = u_i(z_i) - u_i(y_i) \tag{2.7}$$

Proof. *As $r_j \prec_j R_j$, we necessarily have $y_i \succsim_i x_i$ and $z_i \succsim_i y_i$, otherwise we would have $(x_i, R_j, \mathbf{0}_{-ij}) \succ (y_i, r_j, \mathbf{0}_{-ij})$ and $(y_i, R_j, \mathbf{0}_{-ij}) \succ (z_i, r_j, \mathbf{0}_{-ij})$. Therefore, since $x_i \succsim_i \mathbf{0}_i$, we have $z_i \succsim_i y_i \succsim_i x_i \succsim_i \mathbf{0}_i$ and finally since $r_j \prec_j R_j \precsim_j \mathbf{0}_j$, we have $u_j(r_j) < u_j(R_j) \leq 0 \leq u_i(x_i) \leq u_i(y_i) \leq u_i(z_i)$. Thus, $h^U_{w,w'}(x_i, R_j, \mathbf{0}_{-ij}) = u_i(x_i)w(\{i\}) + u_j(R_j)(1 - w'(N \setminus \{j\}))$ and similarly $h^U_{w,w'}(y_i, r_j, \mathbf{0}_{-ij}) = u_i(y_i)w(\{i\}) + u_j(r_j)(1 - w'(N \setminus \{j\}))$. Thus, from the first indifference statement $(x_i, R_j, \mathbf{0}_{-ij}) \sim (y_i, r_j, \mathbf{0}_{-ij})$, we have: $h^U_{w,w'}(x_i, R_j, \mathbf{0}_{-ij}) = h^U_{w,w'}(y_i, r_j, \mathbf{0}_{-ij})$ and therefore $(u_i(y_i) - u_i(x_i))w(\{i\}) = (u_j(R_j) - u_j(r_j))(1 - w'(N \setminus \{j\}))$. Moreover, using the second indifference $(y_i, R_j, \mathbf{0}_{-ij}) \sim (z_i, r_j, \mathbf{0}_{-ij})$, we obtain $(u_i(z_i) - u_i(y_i))w(\{i\}) = (u_j(R_j) - u_j(r_j))(1 - w'(N \setminus \{j\}))$. Then $(u_i(y_i) - u_i(x_i))w(\{i\}) = (u_i(z_i) - u_i(y_i))w(\{i\})$. Finally, since $(\mathbf{1}_i, \mathbf{0}_{-i}) \succ \mathbf{0}$, i.e., $w(\{i\}) > 0$ we obtain Equation 2.7.*

Here also, assuming $(1_i, r_j, \mathbf{0}_{-ij}) \succsim (x_i, R_j, \mathbf{0}_{-ij}) \succsim (0_i, r_j, \mathbf{0}_{-ij})$, the existence of answer y_i is due to restricted solvability. Again, a similar reasoning guarantees the existence of an answer to the second query. Finally, Figure 2.4 (right) represents the two indifference statements in the plan $X_i \times X_j$.

1.2.2 Monotonic Spline Regression

Similarly to Algorithm 2.1, we propose to construct a database of minimal length standard sequences $D = (x_i^\ell, y_i^\ell, z_i^\ell)_{\ell=1}^T$ where for any ℓ, $u_i(x_i^\ell) - u_i(y_i^\ell) = u_i(y_i^\ell) - u_i(z_i^\ell)$. Again, multiplicity is obtained by varying the initial location x_i (below and above the neutral level), the reference dimension j and the mesh (r_j, R_j). Then a monotonic regression based on I-spline functions is performed using database D.

More precisely, we consider here that elements of X_i are real values such that for any $x_i, x_i' \in X_i$, $x_i \succsim_i x_i' \Leftrightarrow x_i \geq x_i'$. If not, elements of X_i could be numerically encoded according to $\succsim_i$. Then, marginal utility u_i is modeled as a linear combination of I-spline basis functions $I_1, \dots, I_m$ defined over $[\mathbf{-1_{\mathit{i}}}; \mathbf{1_{\mathit{i}}}]$ with non-negative coefficients

$\alpha_i = (\alpha_{i,1}, \ldots, \alpha_{i,m})$ as follows:

$$u_{i,\alpha_i}(x_i) = 2\sum_{l=1}^{m} \alpha_{l,i} I_l(x_i) - 1 \tag{2.8}$$

Note that the latter formulation guarantees $u_{i,\alpha}(\mathbf{-1_i}) = -1$ and $u_{i,\alpha}(\mathbf{1_i}) = 1$ if $\sum_{l=1}^{m} \alpha_{l,i} = 1$ since for any $l \in \{1, \ldots, m\}$, $I_i(\mathbf{-1_i}) = 0$, and $I_i(\mathbf{1_i}) = 0$.

Then, using Equation 2.8, the problem of finding the marginal utility that best fits the constraints associated with database D can be formalized as a linear program with the relaxed constraints:

$$\begin{aligned}
&\min_{\alpha_i \in \mathbb{R}^m} \sum_{\ell=1}^{T} (\epsilon_\ell^+ + \epsilon_\ell^-) && (2.9)\\
&\sum_{l=1}^{m} \alpha_{i,l}(2I_l(x_i^\ell) - I_l(y_i^\ell) - I_l(z_i^\ell)) = \epsilon_\ell^+ - \epsilon_\ell^-, \quad \ell = 1, \ldots, T\\
&\sum_{l=1}^{m} \alpha_{i,l} = 1\\
&2\sum_{l=1}^{m} I_l(\mathbf{0}_i)\alpha_{i,l} - 1 = 0 && (2.10)\\
&\epsilon_\ell^+ \geq 0, \epsilon_\ell^- \geq 0, \ell = 1, \ldots, T\\
&\alpha_{l,i} \geq 0, \ l = 1, \ldots, m
\end{aligned}$$

where constraint 2.10 guarantees $u_{i,\alpha_i}(\mathbf{0}_i) = 0$.

A similar linear program can be considered in the case of non restricted solvability where preference statements replace indifference statements (see Appendix A.1). It is sufficient to substitute linear inequalities used to approximate indifference judgments with linear inequality used to approximate preference judgments.

2 Learning Sparse Representations of Capacities

In this section, marginal utility functions are assumed to have been learned beforehand using one of the methods discussed in the previous section. To simplify notations, alternatives are now described by vectors $z = (z_1, \ldots, z_n)$ whose component z_i is the marginal utility w.r.t. the i^{th} viewpoint. We now focus on learning from preference examples the second type of parameter in the CIU model (resp. bi-CIU model): the capacity (resp. two capacities) parameterizing the Choquet integral (resp. bipolar Choquet integral).

In particular, *the objective is to derive sparse representations of capacities (i.e., with few non-null coefficients) to prevent overfitting and enhance the interpretability of* *the CIU model by obtaining a clear view of the most significant subsets of interacting viewpoints.* However, because capacity coefficients increase with set inclusion, achieving meaningful sparse representations is not straightforward. In response to this, we first show the relevance of the Möbius transform for obtaining sparse representations of a capacity.

2.1 The Möbius Transform for Sparse Capacity Representations

For any group of viewpoints $S \subseteq N$, the capacity weight $w(S)$ reflects its global importance, thereby accumulating the importance of all proper subsets of S. More formally, by monotonicity w.r.t. set inclusion of w, $w(S) \geq w(T)$ for any subgroup $T \subseteq S$. Therefore, as soon as a viewpoint $i \in S$ has non-null individual importance, i.e., $w(\{i\}) > 0$, then $w(S) > 0$. Then, capacities are dense weighting systems for which the concept of sparsity is not relevant.

However, the information encoded in the capacity may admit a compact representation using capacity transforms. An interesting transform of a capacity w is its *Möbius transform* m_w (see Equation 1.4). The coefficients $m_w(S)$, called Möbius masses, can be positive or negative and completely characterize w since we have $w(S) = \sum_{T \subseteq S} m_w(T)$. The latter formula highlights coefficients $m_w(T), T \subseteq S$ as positive or negative contributions to the overall importance $w(S)$ of the group S, and shows that Möbius masses add up to 1 since $\sum_{S \subseteq N} m_w(S) = w(N) = 1$. The following proposition establishes that, by design, the Möbius transform of a capacity has always less non-null coefficients than the capacity itself, making it a suitable option for obtaining sparse representations of capacities.

Proposition 2.4. *For any capacity w, we have that $||m_w||_0 \leq ||w||_0$, where $||.||_0$ denotes the ℓ_0-norm, i.e., the number of non-zero coefficients.*

Proof. *Consider a capacity w and its Möbius transform m_w. If $w(S) = 0$ for some $S \subseteq N$, then by monotonicity w.r.t. set inclusion, $w(T) = 0$ for all $T \subseteq S$. Hence, by definition of the Möbius transform (see Equation 1.4), $m_w(S) = \sum_{T \subseteq S} (-1)^{|S \setminus T|} w(T) = 0$. Then $\{T \subseteq N | w(T) = 0\} \subseteq \{T \subseteq N | m_w(T) = 0\}$ and $||w||_0 = 2^n - 1 - |\{T \subseteq N | w(T) = 0\}| \geq 2^n - 1 - |\{T \subseteq N | m_w(T) = 0\}| = ||m_w||_0$.*

Another alternative representation of a capacity w is its *Interaction index* transform I_w (see Equation 1.6). However, the following result shows that the representation of w in terms of interaction I_w may lack of compactness, in particular when w is a *belief function*

[Dempster, 1967, Shafer, 1976], i.e., when m_w is non-negative.

Proposition 2.5. *Let w be a capacity and m_w and I_w its Möbius and interaction index representations respectively. If m_w is non-negative, then $\|I_w\|_0 \geq 2^{t^*} - 1$ where t^* is the largest size of set T such that $m_w(T) > 0$.*

Proof. *The interaction index I_w is linked to m_w by the following equation: $I_w(S) = \sum_{T \supseteq S} \frac{1}{|T|-|S|+1} m_w(T)$ for all $S \subseteq N$ (see Table 1.2). Hence, for any T s.t. $m_w(T) > 0$, we have $I_w(S) > \frac{1}{|T|-|S|+1} m_w(T) > 0$ for all $S \subseteq T$. Let T^* be a subset of maximal cardinality among those such that $m_w(T) > 0$ and denote by t^* its cardinal, then the $2^{t^*} - 1$ subsets of T^* (excluding the empty set) have a positive interaction index. Hence, $\|I_w\|_0 \geq 2^{t^*} - 1$.*

From Proposition 5, if w is a belief function, the number of non-zero coefficients in the interaction index representation increases exponentially with the size of the largest set with non-zero Möbius mass. This is illustrated in the following example.

Example 2.3. *Let w be the capacity such that for any $S \subset N$, $w(S) = \epsilon|S|/n$ for $0 < \epsilon < 1$ and $w(N) = 1$. First, remark that this capacity, which has $2^n - 1$ non-null coefficients, admits a sparse Möbius transform m_w that equals 0 everywhere except for the singletons where $m_w(S) = \epsilon/n$ and for the grand coalition where $m_w(N) = 1 - \epsilon$. Then, by Proposition 5, since N is the largest set with non-null Möbius mass, the interaction index representation of w admits $2^n - 1$ non-null coefficients and thus is as dense as w. The Choquet integral associated with capacity w is the following simple decision model:*

$$h_\epsilon(z) = \frac{\epsilon}{n} \sum_{i=1}^{n} z_i + (1 - \epsilon) \min_{i \in N} \{z_i\} \tag{2.11}$$

This decision model corresponds to an egalitarist attitude in the aggregation (i.e., focusing on the worst marginal value) refined by an utilitarist criterion (i.e, using the sum of marginal utilities) to break ties. In the following, this model is referred to as the ϵ-min model.

Remark 2.4. Belief functions have been particularly studied in the *theory of evidence* (also known as *Dempster-Shafer theory* [Dempster, 1967, Shafer, 1976]) where they are used to represent and reason about uncertainty. In this framework, the Möbius transform is regarded as a probability distribution (or *belief mass distribution*) over the subsets of N, and subsets with positive mass, i.e., such that $m_w(S) > 0$, are referred to as a *focal elements* (see also [Grabisch, 2016c]).

Hence, the Möbius representation appears better suited to provide a sparse representation of the capacity than the interaction index representation (or the capacity itself). Below, we recall the Choquet integral formulation in terms of Möbius masses :

$$C_w(z) = \sum_{S \subseteq N} m_w(S) \min_{i \in S}\{z_i\} \tag{2.12}$$

This formulation highlights an additional benefit of sparse Möbius transforms by suggesting that $C_w(z)$ admits a very simple form when m_w is sparse, composed of conjunctive or disjunctive terms $m_w(S)\min_{i\in S}\{z_i\}$ (depending on the sign of the Möbius mass $m_w(S)$). For instance, the ϵ-min model (see Equation 2.11) only involves a unique conjunctive term that is the minimum of all marginal utilities and a linear term, since the unique non-null masses are located on N and the singletons. Another example is given below for decision-making under uncertainty:

Example 2.4. *Let us consider again the Ellsberg's urn example (see Example 2.1). Recall that we consider an urn containing 90 balls including 30 red, and 60 blue or yellow balls in unknown proportion and four bets: on the one hand x (resp. y) yielding 100\$ if the drawn ball is red (resp. blue), and on the other hand z (resp. t) yielding 100\$ if the drawn ball is not blue (resp. not red). Here $N = \{R, B, Y\}$ for red, blue, yellow, and the acts under consideration are $x = (100, 0, 0)$, $y = (0, 100, 0)$, $z = (100, 0, 100)$ and $t = (0, 100, 100)$.*

As evidenced in Example 2.1, preferences $x \succsim y$ and $z \prec t$ are representable by a Choquet integral associated with the capacity yielding the lower possible probability of the event according to our knowledge of the urn content, i.e., $w(\{R\}) = 1/3, w(\{B\}) = w(\{Y\}) = 0$, $w(\{R, B\}) = w(\{R, Y\}) = 1/3$, $w(\{B, Y\}) = 2/3$ and $w(\{R, B, Y\}) = 1$. Then, we remark that this capacity admits a sparse Möbius transform which equals everywhere 0 except that $m(\{R\}) = 1/3$ and $m(\{B, Y\}) = 2/3$, yielding a simple Choquet integral formulation fitting the observed preferences: $C_w(z_1, z_2, z_3) = z_1/3 + 2\min\{z_2, z_3\}/3$. Note that w is a belief function and the events $\{R\}$ and $\{B, Y\}$ are the focal elements (see Remark 2.4).

Sparse Möbius transforms are commonly employed for controlling the complexity of the Choquet integral [Grabisch and Labreuche, 2010, Hüllermeier and Tehrani, 2013]. Indeed Möbius masses are frequently required to vanish for all subsets of viewpoints larger than a given $k < n$. In this case, the resulting capacity is said to be k-additive [Grabisch, 1997b]. For instance, when the capacity is 1-additive then all Möbius masses are null except for some singletons (at least one) where they are positive due to monotonicity. In this case, Equation 2.12 shows that the Choquet integral boils down to a simple weighted sum.

Considering only 2-additive capacities is a standard option to allow pairwise interactions while keeping a sparse model. One may also wish to relax 2-additivity for k-additivity ($2 < k < n$) with the aim of finding a better tradeoff between sparsity and expressivity. However, *reasoning about sparsity in terms of k-additivity is a drastic reduction that may significantly impact our ability to fit preference data with relevant Choquet integral models*. It may indeed happen that very sparse but still n-additive capacities are necessary to describe preference data, as shown hereafter:

Example 2.5. *Let us consider again the ϵ-min model h_ϵ (see Equation 2.11). Recall that h_ϵ is an instance of the Choquet integral associated with a Möbius transform m_w that is everywhere 0 excepted on singletons and on N ($m_w(\{i\}) = \epsilon/n$ for all $i \in N$ and $m_w(N) = 1-\epsilon$). As the most important Möbius mass is located on the grand coalition N, preferences induced by h_ϵ could not be properly described by any k-additive capacity with $k < n$ despite the fact that m_w can be closely approximated by a sparse Möbius transform involving a single non-null Möbius mass (attached to N).*

This shows that new approaches are needed to find sparse representations of capacities that best fit observed preferences, regardless of k-additivity. In the following, we propose a general approach to learn sparse Möbius representations of capacities allowing to derive simple instances of the (bi-)CIU model that best fit preference data.

2.2 Sparse Möbius Learning with ℓ_1-regularization

Our objective is to find a capacity w with as many zero Möbius masses as possible and such that C_w accurately describes a given set of preference examples. To this end, we formulate a regularized empirical risk minimization (RERM) problem (see Definition 1.21) using ℓ_1-regularization (see Section 3.1.3) over the Möbius transform of w.

More precisely, the aim here is to fit preference examples with a simple model while exploiting the descriptive advantage of the Choquet integral over the standard weighted sum model. The basic model is therefore the weighted sum including all singletons, and then the objective is to include as few interaction terms as possible, through a ℓ_1-regularization of the Mobius mass vector focusing only on the Mobius masses of subsets of size larger than 1. Such a regularization allows exploring the trade-off between model simplicity and error on the preference examples by progressively increasing the weight of the regularization term, until obtaining a linear model.

Besides, an important question arises: *how significant is the selection of interaction terms obtained with ℓ_1-regularization?* Indeed, the Möbius mass selection performed by the ℓ_1-regularization might be impacted by the structural dependence that exists between the quantities of type $\min_{i \in S}\{z_i\}, S \subseteq N$, involved in C_w (Equation 2.12). In particular, by taking a statistical view of the learning problem, we will see that the correlation between these quantities can harm the ability of the ℓ_1-regularization to select the interaction factors properly. We thus propose to use a standard approach to correct this issue (known as *adaptive ℓ_1-regularization* [Zou, 2006]) that consists of using a weighted ℓ_1-regularization with weights derived from preference data. For the sake of clarity, we first present the standard version of the ℓ_1-regularized learning problem (Subsection 2.2.1) and shed light on its limitations in selecting the relevant Möbius masses. A more sophisticated version using a weighting system is then presented in Subsection 2.2.2.

2.2.1 ℓ_1-regularization on the Möbius Transform

Let us consider a set of preference statements $\{(z^\ell, z'^\ell)\}_{\ell \in P}$ and a set of indifference statements $\{(z^\ell, z'^\ell)\}_{\ell \in I}$ over pairs of marginal utility vectors $(z^\ell, z'^\ell) \in [a, b]^2$, where for any $\ell \in P$, $z^\ell \succsim z'^\ell$ and for any $\ell \in I$, $z^\ell \sim z'^\ell$. The error made with C_w on example (z^ℓ, z'^ℓ) is computed with the *pref-hinge loss* (see Definition 1.28) with $\delta = 0$, i.e.:

$$l(C_w(z^\ell), C_w(z'^\ell)) = \begin{cases} |C_w(z^\ell) - C_w(z'^\ell)| & \text{if } \ell \in I \\ (C_w(z) - C_w(z'))^- & \text{if } \ell \in P \end{cases} \tag{2.13}$$

Then, the RERM problem with ℓ_1-regularization over the Möbius masses vector restricted to the interaction subsets (subsets of size strictly larger than 1), is formulated as follows:

$$(\mathcal{P}) \qquad \min_w \sum_{\ell \in I} |C_w(z^\ell) - C_w(z'^\ell)| + \sum_{\ell \in P} (C_w(z^\ell) - C_w(z'^\ell))^- + \lambda \sum_{S \subseteq N, |S|>1} |m_w(S)|$$

$$\textstyle\sum_{T \subseteq S, T \ni i} m_w(T) \geq 0, \quad \forall S \subseteq N, \forall i \in S \tag{2.14}$$

$$\textstyle\sum_{S \subseteq N} m_w(S) = 1 \tag{2.15}$$

where λ is a nonnegative hyper-parameter that controls the level of regularization. Constraints 3.28 and 3.29 respectively ensure the monotonicity of the capacity w.r.t. set inclusion and its normalization. Monotonicity of the capacity can indeed be guaranteed by asking that for any viewpoints coalition $S \subseteq N$, removing a viewpoint $i \in S$ cannot increase the capacity value, i.e., $w(S) \geq w(S \setminus \{i\})$, which translates in terms of m_w by Constraint 3.28 using $w(S) = \sum_{T \subseteq S} m_w(T)$. The latter formula also shows that constraint 3.29 is equivalent to $w(N) = 1$.

2.2.2 Interaction Selection Quality

In the optimization problem $\mathcal{P}$, the ℓ_1-penalty allows sparse representations of capacities to be obtained by shrinking Möbius masses $m_w(S)$ towards zero (the intensity of the shrinkage depending on the level of regularization). Then, a selection of the viewpoints interactions that actually play in the model is performed. As a consequence, it is of prime importance to assess the quality of such a selection. In what follows, by leveraging statistical learning results, and in particular the notion of *variable-selection consistency* and *general sign consistency*, we provide theoretical insights justifying the need for a more sophisticated ℓ_1-regularization to perform qualitative selection of interaction viewpoints.

Capacity learning is a linear regression problem In order to make explicit a possible issue in the interaction selection performed by ℓ_1-regularization, let us consider a special case of Problem $\mathcal{P}$ wherein the database of learning examples is only made of indifference statements with specific pairs of examples $(z^\ell, z'^\ell), \ell \in I$, chosen in such a way that z'^ℓ has a constant marginal utility vector (i.e., $z_i'^\ell = y^\ell$ for all $i \in N$ for some $y^\ell \in \mathbb{R}$). In such a case we have $C_w(z'^\ell) = y^\ell$ whatever the capacity w. Therefore the indifference $z^\ell \sim z'^\ell$ translates into the constraint $C_w(z^\ell) = y^\ell$. Hence, Problem $\mathcal{P}$ boils down to a regression problem with the Choquet integral model C_w using data points $\{(z^\ell, y^\ell)\}_{\ell=1}^t$, where t is the number of examples. Such a dataset could be alternatively obtained by directly collecting global evaluations y^ℓ of marginal value vectors z^ℓ. In this setting, with the monotonicity constraints set aside, the learning problem reduces to the following regression problem:

$$\min_w \sum_{\ell=1}^{t} |C_w(z^\ell) - y^\ell| + \lambda \sum_{S \subseteq N, |S|>1} |m_w(S)| \tag{2.16}$$

This optimization problem falls into the category of *linear regression* problems with ℓ_1-regularization and *absolute loss* (i.e., $\ell(y, \hat{y}) = |\hat{y} - y|$). Indeed, denoting $\beta_j = m_w(S_j)$ and $\phi_j = \min_{i \in S_j}\{z_i\}$ (where S_j is the j^{th} subset of N in the lexicographical order) Equation 2.12 presents C_w as a linear aggregator within a specific feature space of size $d = 2^n - 1$, i.e.:

$$C_w(z) = \sum_{j=1}^{d} \beta_j \phi_j \tag{2.17}$$

Often referred to as LAD-LASSO (where LAD stands for least absolute deviation because of the absolute loss) [Wang et al., 2007, Gao and Huang, 2010], ℓ_1-regularized linear regression with the absolute loss has been extensively studied in the statistical learning literature and, in particular, its properties concerning *variable selection* are now well understood. Here, we propose to leverage the notion of *variable-selection consis-*

tency and *general sign consistency* for assessing the quality of the interaction selection performed by ℓ_1-regularization within the Choquet integral.

Variable-selection consistency A way to assess the quality of coefficient selection in the sparse learning of a linear model[1] $y = \sum_{j=1}^{d} \beta_j \phi_j$ from regression examples $\{\phi^\ell, y^\ell)\}_{\ell=1}^{t}$ is to adopt a statistical view of the learning problem, assuming that the data is such that $y^\ell = \sum_{j=1}^{d} \beta_j^* \phi_j^\ell + \epsilon^\ell, \ell = 1, \ldots, t$, where $(\epsilon^1, \ldots, \epsilon^t)$ is a vector of i.i.d. centered random variables, and $\beta^* \in \mathbb{R}^d$ embodies a *ground truth model*. This ground truth model is further assumed to be sparse in the sense that some of its components are zero. Non-null components indices are listed in $A_1 = \{j | \beta_j^* \neq 0\}$ while null components indices are listed in A_2, i.e., $A_2 = \{j | \beta_j^* = 0\}$. Finally, without loss of generality, the data is assumed to be centered and normalized i.e., $\frac{1}{t}\sum_{\ell=1}^{t} \phi_j^\ell = 0$ and $\frac{1}{t}\sum_{\ell=1}^{t} (\phi_j^\ell)^2 = 1$, $j = 1, \ldots, d$.

Let $\hat{\beta}_t(\lambda)$ be a model learned from such data using a RERM problem with a sparsity -inducing regularization with hyper-parameter λ. Then, in this setting, assessing the variable selection performed by the learning method amounts to asking the question: "*is $\hat{\beta}_t(\lambda)$ guaranteed to recover with high probability the set A_1 of relevant variables if provided with a large amount of data t and the proper amount of regularization λ?*". If the answer is yes, then the learning method is said to be *variable-selection consistent* [Zhao and Yu, 2006, Hastie et al., 2015b]. A stronger property is the *general sign consistency*, which guarantees that not only the coefficient of A_1 are recovered but the sign of the estimated coefficients coincides with those of the ground truth model [Zhao and Yu, 2006, Gao and Huang, 2010]. This property can be formally formulated as follows:

Definition 2.5 (general sign consistency). *An estimator $\hat{\beta}_t(\lambda)$ is general sign consistent if:*

$$\lim_{t \to \infty} \mathbb{P}(\exists \lambda \geq 0, \operatorname{sign}(\hat{\beta}_t(\lambda) = \operatorname{sign}(\beta^*)) = 1$$

where for any vector β, $\operatorname{sign}(\beta)$ refers to its sign vector, i.e., $\operatorname{sign}(\beta)_j = 1$ if $\beta_j > 0$, $\operatorname{sign}(\beta)_j = -1$ if $\beta_j < 0$ and $\operatorname{sign}(\beta)_j = 0$ otherwise.

In words, an estimator $\hat{\beta}_t(\lambda)$ is general sign consistent if the probability of the existence of a λ value for which it correctly affects signs to coefficients goes towards 1 as the number of examples approaches infinity. The question of whether LAD-LASSO is *general sign consistent* has been addressed and *negative results have been provided in the setting where features are highly correlated with each other* [Gao and Huang, 2010].

[1] Here ϕ_j represents a general feature in a linear regression problem

The irrepresentable condition It is known that, under mild assumptions, a necessary condition for LAD-LASSO to be general sign consistent is the (weak) *irrepresentable condition* (IC) [Gao and Huang, 2010]. More precisely, let Φ_t be the design matrix, i.e., the matrix of size $t \times d$ containing the feature observations $((\phi_1^\ell, \dots, \phi_d^\ell))_{\ell=1}^t$. Then, let Φ_t^j be the j^{th} column of Φ_t, and let $\Phi_t^{\mathbf{1}}$ and $\Phi_t^{\mathbf{2}}$ be the submatrices containing the columns indexed by $j \in A_1$ and $j \in A_2$, respectively. Also, let Σ_t denote $\frac{1}{t}\Phi_t^\top \Phi_t$ the sample correlation matrix and assume Σ_t converges to a positive definite matrix Σ when $t \to \infty$. Finally denote by $\Sigma_t^{\mathbf{11}}$ the sub-matrix of Σ_t containing the correlations between variables in A_1, i.e., $\Sigma_t^{\mathbf{11}} = \frac{1}{t}(\Phi_t^{\mathbf{1}})^\top \Phi_t^{\mathbf{1}}$, and denote by $\Sigma_t^{\mathbf{21}}$ the sub-matrix of Σ_t containing the correlations between variables in A_2 and A_1, i.e., $\Sigma_t^{\mathbf{21}} = \frac{1}{t}(\Phi_t^{\mathbf{2}})^\top \Phi_t^{\mathbf{1}}$.

Definition 2.6 (weak irrepresentable condition (IC)). *Assuming $\Sigma_t^{\mathbf{11}}$ is invertible, the (weak) IC condition reads as follows:*

$$|\Sigma_t^{\mathbf{21}}(\Sigma_t^{\mathbf{11}})^{-1}\,\mathrm{sign}(\beta^*_{A_1})| < \mathbf{1} \tag{2.18}$$

where $\mathbf{1} = (1, \dots, 1)$ is a vector of size $|A_2|$ and the inequality holds element-wise.

Such a condition means that when irrelevant variables (A_2 variables) are too correlated with the relevant variables (A_1 variables), LAD-LASSO does not satisfy *general sign consistency*. A way to see it is to observe that Condition 2.18 always holds if A_2 variables are not correlated with A_1 variables (i.e., $\Sigma_t^{\mathbf{21}}$ is null). Alternatively, we can observe that Condition 2.18 for $\mathrm{sign}(\beta^*_{A_1}) = \mathbf{1}$ reduces to $\|(\Phi_t^i)^\top \Phi_t^{\mathbf{1}}((\Phi_t^{\mathbf{1}})^\top \Phi_t^{\mathbf{1}})^{-1}\|_1 < 1$, for any $i \in A_2$. This latter condition is equivalent to saying that if we were regressing an irrelevant variable $i \in A_2$ on all the relevant variables $j \in A_1$(using least square linear regression), the ℓ_1-norm of the obtained coefficient vector should be less than 1. Indeed, recall that the solution of $\arg\min_{\boldsymbol{\beta}} \|\Phi_t^i - \Phi_t^{\mathbf{1}}\boldsymbol{\beta}\|_2^2$ is given by $\hat{\beta} = ((\Phi_t^{\mathbf{1}})^\top \Phi_t^{\mathbf{1}})^{-1}(\Phi_t^{\mathbf{1}})^\top \Phi_t^i = ((\Phi_t^i)^\top \Phi_t^{\mathbf{1}}((\Phi_t^{\mathbf{1}})^\top \Phi_t^{\mathbf{1}})^{-1})^\top$.

Therefore, if an irrelevant variable is too correlated with the variables associated with non-null ground truth coefficients, LAD-LASSO may fail to distinguish it from the latter relevant variable, regardless of the amount of data at hand or the level of regularization applied. Note that a similar result [Zhao and Yu, 2006, Zou, 2006] is also available for least square ℓ_1-penalized linear regression (also known as LASSO regression [**?**Hastie et al., 2015b]).

In the following, we show that these results allow us to highlight potential issues when performing Möbius masses selection with ℓ_1-regularization.

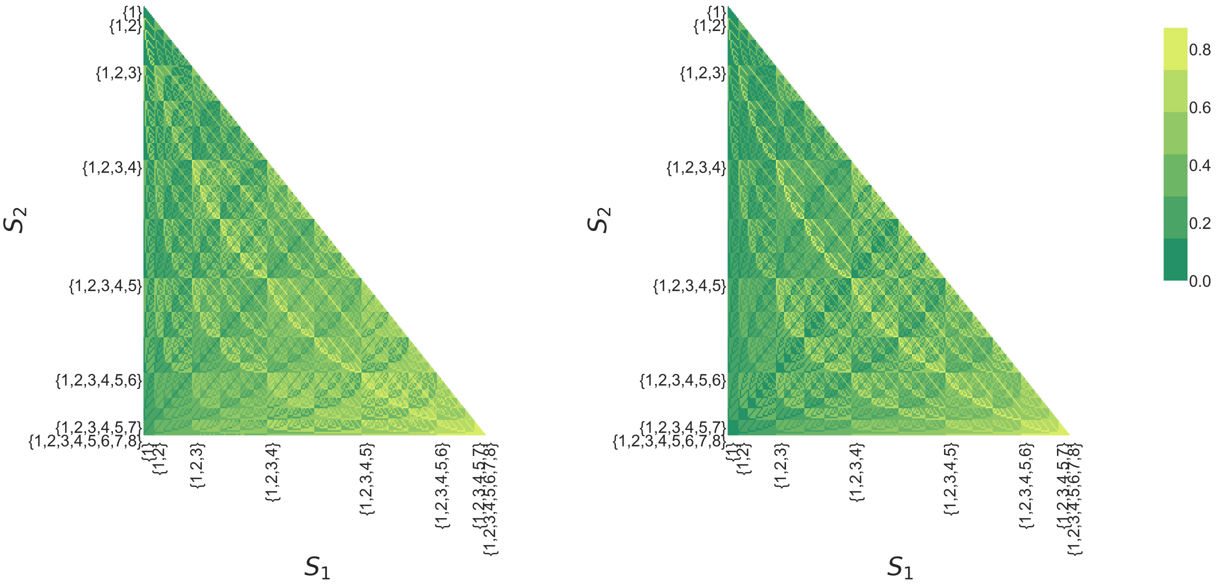


Figure 2.5: Sample correlation between ϕ_{S_1} and ϕ_{S_2} (right) and R ratio (left).

The case of the Choquet integral In the case of the Choquet integral, the feature space is endowed with a very specific correlation structure. Indeed, features are indexed over subsets $S \subseteq N$ and for any pair of criteria coalition $S_1, S_2 \subseteq N$ such that $S_1 \cap S_2 \neq \emptyset$, $\phi_{S_1} = \min_{i \in S_1}\{z_i\}$ and $\phi_{S_2} = \min_{i \in S_2}\{z_i\}$ are obviously statistically correlated due to the overlapping of the coalitions. Intuitively, the correlation is all the more important that the cardinal of the intersection is close to the cardinal of the union. This is well illustrated in Figure 2.5 that compares the ratio $R = |S_1 \cap S_2|/|S_1 \cup S_2|$ for any $S_1, S_2 \subseteq N$ (right handside) and the empirical correlation between $\min_{i \in S_1}\{z_i\}$ and $\min_{i \in S_2}\{z_i\}$ (left handside) when utilities $z_i, i = 1, \dots, n$ are independent random variables distributed according to a uniform distribution within $[0, 1]$. The number of criteria n is taken equal to 8 and for any $i \in N$, i.i.d. utility samples $(z_i^\ell)_{\ell=1}^t$ of size $t = 1000$ are simulated to compute the empirical correlations.

In Example 2.6, we show that this correlation structure undermines the respect of Condition 2.18, and thus the ability of LAD-LASSO to recover a sparse ground truth model.

Example 2.6. *Let us consider the ϵ-min CIU model (see Definition 2.11) for $n = 3$, i.e.:*

$$y = \frac{\epsilon}{3}(z_1 + z_2 + z_3) + (1 - \epsilon) \min_{i \in N}\{z_i\}$$

This model can be identified to the model $y = \sum_{S \subseteq N} \phi_S \beta_S^*$ with $\beta_S^* = \epsilon/3$ for S such that $|S| = 1$, $\beta_N^* = 1 - \epsilon$, and $\beta_S^* = 0$ otherwise. Then the indices of the non-null coefficients in the ground truth model are $A_1 = \{\{1\}, \{2\}, \{3\}, N\}$, $A_2 = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$

and $\operatorname{sign}(\beta^*_{A_1}) = \mathbf{1}$*. Suppose that the utilities* $z_i, i = 1, 2, 3$ *are independent random variables distributed according to a uniform distribution within* $[0, 1]$*. Let* $\rho^{s_{12}}_{s_1,s_2}$ *be the correlation between* ϕ_{S_1} *and* ϕ_{S_2} *for any* $S_1, S_2 \subseteq N$ *such that* $|S_1| = s_1, |S_2| = s_2$*,* $|S_1 \cap S_2| = s_{12}$*, i.e.,* $\rho^{s_{12}}_{s_1,s_2} = \operatorname{Cov}(\phi_{S_1}, \phi_{S_2})/\sqrt{\operatorname{Var}(\phi_{S_1}) \operatorname{Var}(\phi_{S_1})}$*. Then, the correlation matrix* $\Sigma^{\mathbf{11}}$ *and* $\Sigma^{\mathbf{21}}$ *are given by:*

$$\Sigma^{\mathbf{11}} = \begin{pmatrix} 1 & 0 & 0 & \rho^1_{1,3} \\ 0 & 1 & 0 & \rho^1_{1,3} \\ 0 & 0 & 1 & \rho^1_{1,3} \\ \rho^1_{1,3} & \rho^1_{1,3} & \rho^1_{1,3} & 1 \end{pmatrix}, \qquad \Sigma^{\mathbf{21}} = \begin{pmatrix} \rho^1_{1,2} & \rho^1_{1,2} & 0 & \rho^2_{2,3} \\ \rho^1_{1,2} & 0 & \rho^1_{1,2} & \rho^2_{2,3} \\ 0 & \rho^1_{1,2} & \rho^1_{1,2} & \rho^2_{2,3} \end{pmatrix}$$

Then if $1 - 3(\rho^1_{1,3})^2 \neq 0$*, Condition 2.18 boils down to (see Proposition 7.9 in Appendix A.2):*

$$|2\rho^1_{1,2}(1 - \rho^1_{1,3}) + \rho^2_{2,3}(1 - 3\rho^1_{1,3})| < |1 - 3(\rho^1_{1,3})^2| \tag{2.19}$$

Using analytical formulas for $\operatorname{Cov}(\phi_{B_1}, \phi_{B_2})$ *provided in Proposition 7.10 in Appendix A.2, we have* $\rho^1_{1,2} = \frac{\sqrt{3}}{2\sqrt{2}}, \rho^1_{1,3} = \frac{1}{\sqrt{5}}, \rho^2_{2,3} = \frac{4}{\sqrt{30}}$*. Therefore, Condition 2.19 is equivalent to* $\frac{3+\sqrt{5}}{5\sqrt{6}} < \frac{2}{5}$*, which is false, and thus Condition 2.18 is violated.*

The violation of Condition 2.18 in Example 2.6 suggests some weaknesses of the ℓ_1-regularizations in terms of viewpoints interaction selection. In order to circumvent this issue, we investigate the benefit of an adaptive ℓ_1-penalty, i.e., a weighted ℓ_1-penalty with data-dependent weights.

Remark 2.5. In the previous analysis marginal utilities $(z_1, \ldots, z_n)$ have been assumed to be statistically independent to asses the sole impact of the structural dependence between features $\phi_S, S \subseteq N$ due to set inclusions. Taking into account the correlation between marginal utilities would only increase correlations between relevant and irrelevant features and undermine the satisfaction of Condition 2.18. Additionally, note that the correlation between features $\phi_S, S \subseteq N$ has also been highlighted as a potential source of numerical instabilities when learning the Choquet integral [Tehrani and Ahrens, 2017].

2.2.3 Learning Möbius Masses with Adaptive ℓ_1-regularization

The adaptive ℓ_1-regularization is a regularization of the form $\sum_j \lambda_j |\beta_j|$ where the weights λ_j are data-dependent and adapted to each coefficient β_j, implying a two-stage algorithm where the first step is the weights computation. It has been introduced to correct LASSO and LAD-LASSO and guarantee better variable selection properties [Zou, 2006, van de Geer, 2010, Xu and Ying, 2010, Zheng et al., 2017, Wu et al., 2022]. The underlying idea is that if the weights λ_j contain information about the relative relevance of the variables, it allows for less regularization on relevant variables and more regularization on less relevant variables, thereby mitigating the blurriness between relevant and irrelevant variables due to correlation.

In particular, when the weights are the reciprocals of absolute values of the coefficients obtained with a ridge regression (i.e., ℓ_2-regularized least square regression) the adaptive LASSO is known to be variable-selection consistent [Zou, 2006]. Therefore, we propose to use this two-stage penalty in the learning of capacities from sets of preference and indifference examples $\{(z^\ell, z'^\ell)\}_{\ell\in P}$ and $\{(z^\ell, z'^\ell)\}_{\ell\in I}$. It yields the following learning problem:

$$(\mathcal{P}') \qquad \min_w \sum_{\ell\in I} |C_w(z^\ell) - C_w(z'^\ell)| + \sum_{\ell\in P} (C_w(z^\ell) - C_w(z'^\ell))^- + \sum_{S\subseteq N, |S|>1} \lambda_S |m_w(S)|$$
$$\text{s.t. } 3.28, 3.29$$

The weights used in the regularization are defined by $\lambda_S = \lambda/(|\hat{m}_w(S)| + \epsilon)$ for any $S \subseteq N$ ($\epsilon > 0$ being introduced to avoid numerical instabilities), where $\hat{m}_w$ is the optimal solution of the ℓ_2-regularized preference learning problem, given below:

$$(\mathcal{P}'_0) \qquad \min_w \sum_{\ell\in I} |C_w(z^\ell) - C_w(z'^\ell)| + \sum_{\ell\in P} (C_w(z^\ell) - C_w(z'^\ell))^- + \lambda_0 \|m_w\|_2^2$$
$$\text{s.t. } 3.28, 3.29$$

Note that λ_0 is an additional nonnegative regularization hyper-parameter controlling the level of ℓ_2-regularization in the first step.

Then, for both problems $\mathcal{P}'_0$ and $\mathcal{P}'$, we introduce auxiliary variables $\epsilon^+_\ell, \epsilon^-_\ell$ (resp. ϵ_ℓ) to linearize the indifference violations $|C_w(z^\ell) - C_w(z'^\ell)|$ (resp. preference violations $(C_w(z^\ell) - C_w(z'^\ell))^-$), as well as variables a_S, b_S to linearize the quantities $|m_w(S)|$ involved in the objective function of problem $\mathcal{P}'$ (see Remark 2.2 for detail on linearization of absolute values). Additionally, since $C_w(z^\ell) = \sum_{S\subseteq N} m_w(S) \min_{i\in S}\{z^\ell_i\}$ by Equation 2.12, we have $C_w(z^\ell) - C_w(z'^\ell) = \sum_{S\subseteq N} m_w(S)\Delta^\ell_S$ with $\Delta^\ell_S = \min_{i\in S}\{z^\ell_i\} - \min_{i\in S}\{z'^\ell_i\}$.

Therefore, problem $\mathcal{P}'$ reduces to the following linear program:

$$\min \sum_{\ell \in I}(\epsilon_\ell^+ + \epsilon_\ell^-) + \sum_{\ell \in P} \epsilon_\ell + \sum_{S \subseteq N, |S|>1} \lambda_S(a_S + b_S)$$

$$\textstyle\sum_{S \subseteq N}(a_S - b_S)\Delta_S^\ell + \epsilon_\ell^+ - \epsilon_\ell^- = 0, \quad \ell \in I \tag{2.20}$$

$$\textstyle\sum_{S \subseteq N}(a_S - b_S)\Delta_S^\ell + \epsilon_\ell \geq 0, \quad \ell \in P \tag{2.21}$$

$$\textstyle\sum_{T \subseteq S, T \ni i} a_T - b_T \geq 0, \quad \forall S \subseteq N, \forall i \in S \tag{2.22}$$

$$\textstyle\sum_{S \subseteq N} a_S - b_S = 1 \tag{2.23}$$

$$\epsilon_\ell^+, \epsilon_\ell^-, \epsilon_\ell, a_S, b_S \geq 0$$

Equations 2.20 and 2.21 correspond to the constraints induced by the indifference and preference examples, while constraints 2.22 and 2.23 respectively impose the monotonicity and the normalization of the capacity. Note that Möbius masses are recovered by $m_w(S) = a_S - b_S$ and that weights λ_S need to be priorly computed by solving $\mathcal{P}'_0$, which reduces to a quadratic program after linearization of the indifference and preference violations.

In order to derive a similar optimization problem for the learning of model bi-CIU, let us reformulate $BC_{w,w'}$ from the Möbius transforms of capacities w and w'. From Definition 1.14 and Equation 2.12, if marginal utilities are valued in $[-1, 1]$ where 0 is the neutral level, we have:

$$\begin{aligned} BC_{w,w'}(z) &= \sum_{S \subseteq N} m_w(S) \min_{i \in S}\{z_i^+\} + \sum_{B \subseteq N} m_{w'}(S) \min_{i \in S}\{-z_i^-\} \\ &= \sum_{S \subseteq N} m_w(S) \min_{i \in S}\{z_i^+\} - \sum_{S \subseteq N} m_{w'}(S) \max_{i \in S}\{z_i^-\} \end{aligned} \tag{2.24}$$

Using Equation 2.24, we formulate the problem of learning sparse representations of the capacities w and w' in bi-CIU as follows:

$$\min \sum_{\ell \in I}(\epsilon_\ell^+ + \epsilon_\ell^-) + \sum_{\ell \in P} \epsilon_\ell + \sum_{S \subseteq N, |S|>1} (\lambda_S^w(a_S + b_S) + \lambda_S^{w'}(c_S + e_S))$$

$$\textstyle\sum_{S \subseteq N}((a_S - b_S)\Delta_S^\ell - (c_S - e_S)\nabla_S^\ell) + \epsilon_\ell^+ - \epsilon_\ell^- = 0, \quad \ell \in I$$

$$\textstyle\sum_{S \subseteq N}((a_S - b_S)\Delta_S^\ell - (c_S - e_S)\nabla_S^\ell) + \epsilon_\ell \geq 0, \quad \ell \in P$$

$$\textstyle\sum_{T \subseteq S, T \ni i}(a_T - b_T) \geq 0, \quad \forall S \subseteq N, \forall i \in S$$

$$\textstyle\sum_{T \subseteq S, T \ni i}(c_T - d_T) \geq 0, \quad \forall S \subseteq N, \forall i \in S$$

$$\textstyle\sum_{j=1}^d (a_S - b_S) = 1$$

$$\textstyle\sum_{j=1}^d (c_S - e_S) = 1$$

$$\epsilon_S^+, \epsilon_S^-, \epsilon_\ell, a_S, b_S, c_S, e_S \geq 0$$

where $\Delta_S^\ell = \min_{i\in S}\{(z_i^\ell)^+\} - \min_{i\in S}\{(z_i'^\ell)^+\}$ and $\nabla_S^\ell = \max_{i\in S}\{(z_i^\ell)^-\} - \max_{i\in S}\{(z_i'^\ell)^-\}$. The weights $(\lambda_S^w, \lambda_S^{w'})$ are computed beforehand with a quadratic program similar to $\mathcal{P}_0'$ but using a double ℓ_2-regularization $\lambda_0(\|m_w\|_2^2 + \|m_{w'}\|_2^2)$.

Note that another weighting system based on the cardinality of factors has been used in *Choquistic regression* problems to favor the selection of small-size factors [Tehrani and Hüllermeier, 2013]. However, this choice may prevent to recover preference systems where large coalitions are essential. For instance, this is the case of the ϵ-min model (see Equation 2.11). This is also the case of the so-called Hurwicz model [Hurwicz, 1951] based on a convex combination of min and max factors taken on the grand coalition N, i.e., $h_\alpha(z) = \alpha \min_{i\in N}\{z_i\} + (1-\alpha)\max_{i\in N}\{z_i\}$ for any $\alpha \in [0,1]$.

Remark 2.6 (Elastic Net). Another possible variant of standard ℓ_1-regularization in the context of correlated variables is the *Elastic Net* [Zou and Hastie, 2005]. This penalty is defined as a convex combination of ℓ_1 and ℓ_2 penalties: $\lambda\|m_w\|_1 + \lambda_0\|m_w\|_2^2$. However, this method tends to jointly select correlated features with uniformed coefficient values (grouping effect) as observed in [Zou and Hastie, 2005]. This property is not desirable in our context. For instance, in the ϵ-min model (see Equation 2.11), the importance of $m_w(N)$ in the ground truth model may thwart the elimination of sets having a large intersection with N. This is confirmed by our tests. In Section 3, we will show that adaptive ℓ_1-penalty significantly outperforms both the standard ℓ_1-regularization and the Elastic Net penalty.

3 Numerical Tests

In this section we show the results of numerical experiments on *synthetic* and *real-world* preference data and we illustrate the advantage of our approach over some baseline methods. All tests are conducted on a 2.8 GHz Intel Core i7 processor with 16GB RAM and optimization tasks are performed using the mathematical programming solver Gurobi (version 9.1.2).

3.1 Synthetic data

Model generation Random CIU (or bi-CIU) models with sparse Möbius representation are created through the generation of n (or one for the DMU setting) marginal utility functions u_i and a capacity (or two for bi-CIU) admitting a sparse Möbius representation.

- Marginal utility functions are modeled as non-decreasing splines defined over consequence sets of the form $X_i = [0; M]$ using a basis of I-spline functions of size $m = 10$ associated with a regular subdivision of $[0; M]$ (see Section 1.1.1 for more details on I-spline functions). Then, random marginal utility functions are generated by uniformly drawing I-spline basis coefficients α in the m-dimensional simplex. Note that we use here cubic I-splines ($k = 3$) because they have matching first and second derivatives while preserving a local influence of every component.

- Sparse Möbius masses are first generated without requiring for monotonicity by generating sparse real-valued vectors of size $2^n - 1$ with components summing to 1. Then, m_w or $(m_w, m_{w'})$ are taken as the Möbius representations of the closest (in the sense of the ℓ_1-norm) monotonic capacities to these vectors (obtained by linear programming).

Remark 2.7. Due to monotonicity constraints, the uniform generation of capacities (not necessarily with sparse Möbius transform) is recognized as a difficult task for which tractable algorithms are still needed [Havens and Pinar, 2017, Sun et al., 2023, Grabisch et al., 2023].

For a given random CIU (resp. bi-CIU) model h_w^U (resp. $h_{w,w'}^U$), we generate learning databases for learning the marginal utility functions and the capacity as detailed below.

Data generation for learning the marginal utility functions For learning the marginal utility functions, we simulate Q-queries and their answers to construct databases D of the form $\{(o_1^\ell, o_2^\ell, o_3^\ell)\}_{\ell=1}^T$. More precisely, Q-queries are randomly generated by random draws of their parameters, such as the initial consequence o_1 and the reference consequences R, r (see Proposition 2.1 and Proposition 2.2 respectively for the DMU and MADM setting). Then, answers to these queries are obtained by simulating a DM answering according to h_w^U. For instance, in the DMU setting, for a Q-query $Q_S(o_1|r, R)$, we solve the equation $h_w^U(\bar{o}_2 S \bar{R}) = h_w^U(\bar{o}_1 S \bar{r})$ and disturb the answer with some random uniform noise $\epsilon \in [-\epsilon_{max}, \epsilon_{max}]$, i.e., $o_2 = u^{-1}(u(o_1) + [u(R) - u(r)]w(\bar{S})/(w(\bar{S}) - 1) + \epsilon)$. Unless specified, the noise level $\epsilon_{max} = 0.05$ is used in the following.

Data generation for learning the Möbius transform of the capacity For the capacity learning, we construct a database $D = \{(z^\ell, z'^\ell)\}_{\ell \in P \cup I}$ of preference and indifference statements over marginal value vectors compatible with h_w^U (or $h_{w,w'}^U$). For this, pairs (z^ℓ, z'^ℓ) are drawn uniformly within $[0,1]^n$ (or $[-1,1]^n$ in the case of bi-CIU). In order to introduce noise, each pair example is associated with a preference statement $z^\ell \succsim z'^\ell$ (i.e., $\ell \in P$) if $(h_w^U(z^\ell) + \sigma_z^\ell) - (h_w^U(z'^\ell) + \sigma_{z'}) \geq \sigma$ and $z^\ell \sim z'^\ell$ (i.e., $\ell \in I$) if $|(h_w^U(z^\ell) + \sigma_z^\ell) - (h_w^U(z^\ell) + \sigma_{z'}^\ell)| < \sigma$, where $\sigma_z^\ell, \sigma_{z'}^\ell$ are noise values uniformly drawn within an interval $[-\sigma, \sigma]$ (for CIU). Unless specified, the noise level $\sigma = 0.03$ is used in the following.

n	5	6	8
training set size	100	120	250

Table 2.1: Size of the training preference dataset w.r.t. the number of viewpoints n.

This process is used to generate training datasets D of size $|P| + |I|$ which we vary in our experiments depending on the number of viewpoints n following Table 2.1. Note also that preference and indifference examples are in equal proportions. We also generate test datasets (not used for the learning) to evaluate the generalizing performances of the learned models. Test sets are always of size $|P| + |I| = 1000$. The generalizing performance of any learned model h, is evaluated as the empirical error $R_{emp}(h|D)$ (see Definition 1.20) with the *pref-hinge loss* (see Equation 2.13) on some test dataset D and is referred to as the *test error* in the following.

3.2 Learning marginal utility functions

In this section, we conduct numerical tests to evaluate the methods proposed in Section 1 for learning marginal utility functions within the CIU (or bi-CIU) model. We first show numerical results for the learning method in the DMU setting (see Section 1.1.1), and then for the MADM setting (see Section 1.2.1). As we use synthetic data, the learned marginal utility functions u_{α^*} (see Equation 2.3) are evaluated according to their distance to the ground truth marginal utility functions u used to generate the synthetic data. Referred to as *ground truth distance* and denoted by $d(u_{\alpha^*}, u)$, this distance is computed as the average absolute difference between both functions on a discretization of the consequence set $[0, M]$.

Learning the utility function in DMU First, a random CIU model is generated with a unique marginal utility function u, and then Q-queries and their (noisy) answers according to this model are generated using the generation procedure described in Section 3.1. Following the incremental procedure described in Algorithm 2.1, we increase the learning database D until ρ is sufficiently reduced. The result of the learning process is presented in Figure 2.6 for increasing size of learning database $T = 4, 16, 32$ (from left to right). To illustrate the level of uncertainty associated with the learned function (red plain line), the latter is displayed with the upper bound $\max_{\alpha \in V_\delta(z^*)} u_\alpha(o)$ and the lower bound $\min_{\alpha \in V_\delta(z^*)} u_\alpha(o)$ (blue dotted line) introduced in Subsection 1.1.2. The ground truth utility function is represented in plain black. In this instance, u is already well estimated with tight bounds for $T = 32$.

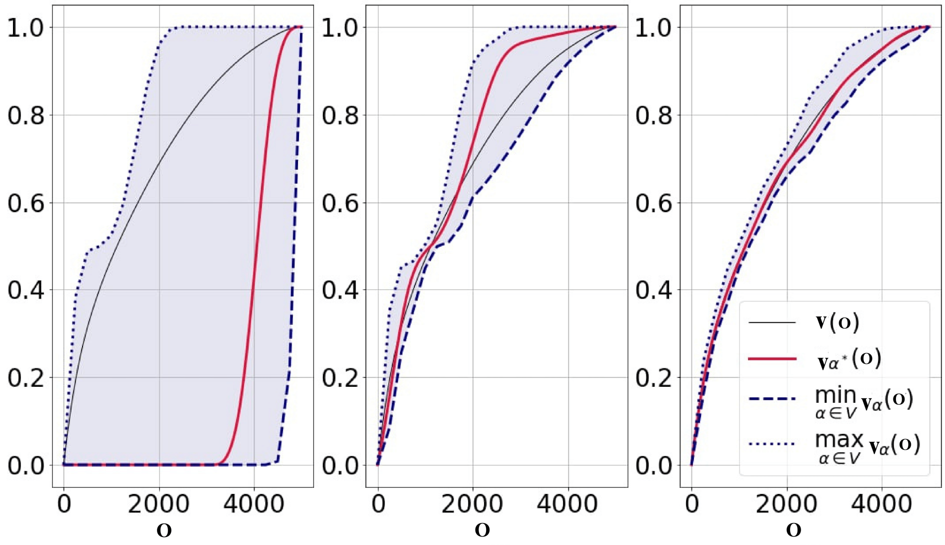


Figure 2.6: Identification of the utility function u_α for $T = 4, 16, 32$ (left to right).

Then, this experiment has been conducted for 1000 random CIU models. In Table 2.2, we show, as the number T of learning examples increases, the decrease of ρ and the average ground truth distance of the learned function. Again, we observe that after 32 examples the utility function is correctly recovered as the ground truth distance is vanishing in average over the 1000 simulations.

	T = 4	T = 16	T = 32
ρ	0.687	0.124	0.072
$d(u, u_{\alpha^*})$	0.354	0.024	0.004

Table 2.2: Average ρ and $d(u, u_{\alpha^*})$ w.r.t the number of constraints T.

Learning the marginal utility functions in MADM We now illustrate the method for learning the marginal utility functions within CIU in the MADM setting (see Section 1.2.1). In these tests, the objective is to compare our method to the elicitation method relying on standard sequences in terms of noise robustness. For the latter method, the final function is taken as a cubic I-spline interpolation of the points obtained with the standard sequence. Below, we illustrate the learning process for any $i \in N$

First, a random bi-CIU model is generated and then Q-queries and their (noisy) answers according to this model are generated using the generation procedure described in Section 3.1. Figure 2.7 displays the learned marginal utility functions for our method (red dashed line) and the standard sequence method (green and blue points), along with the ground truth u_i (black plain line). On the left, the estimation provided by our method perfectly matches the ground truth while on the right the estimation of the standard sequence clearly suffers from noise distortion. We conducted the same experiment on 10 random bi-CIU models, and obtained an average ground truth distance of 0.084 ± 0.052 for the standard sequence method and of 0.022 ± 0.008 for our approach.

Additionally, on Figure 2.8 we represent the ground truth distance (i.e., $d(u, u_{\alpha^*})$) in average over 10 simulations (with standard errors) for both methods, as a function of the number of queries asked, for varying noise in the query answers. More precisely, Figure 2.8 shows the case $\epsilon_{max} = 0.05$ on the left and $\epsilon_{max} = 0.1$ on the right, where the results of the standard sequence method are in plain black and our results are in dotted pink. The test confirms that long standard sequences constructed by chaining answers to preference queries lead to very poor results. Also, the difference between both graphs shows the impact of the increase of noise intensity on the estimation quality for both method. However, one can see that regardless the level of noise, contrarily to the standard sequence method, our approach converges to a vanishing ground truth distance. It thus appears to be a more robust approach.

3.3 Learning Sparse Möbius Representations of Capacities

In this section, we first illustrate the process of learning a sparse Möbius representation of the capacity in the specific case where data is generated with the ϵ-min Choquet integral instance (see Equation 2.11). Then, with other toy examples, we illustrate the benefit of adaptive ℓ_1-regularization in terms of viewpoint interaction selection. Finally, we proceed to experiments demonstrating the benefits of our approach in the general case of sparse synthetic data and real-world preference data.

For all methods, the regularization hyper-parameter λ is chosen by *cross-validation* with the *one-standard-error rule*. Using a set of pre-selected λ values, cross-validation consists of assessing the generalizing performances attached to each λ value by cutting

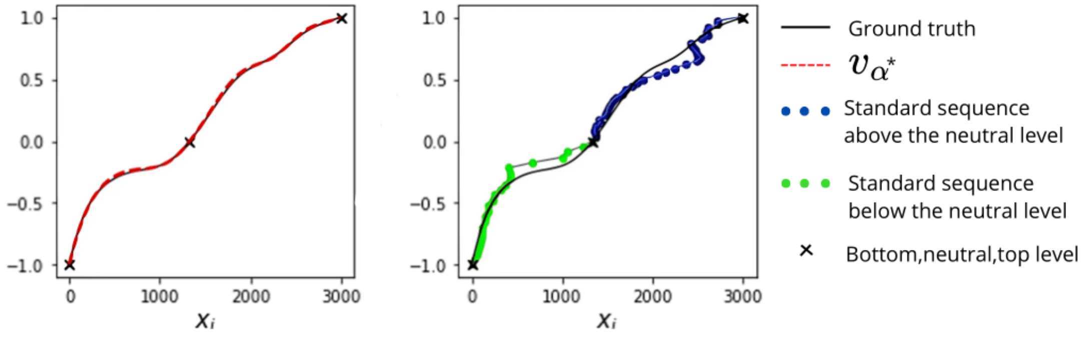


Figure 2.7: Learned function with our method (left) and standard sequences (right).

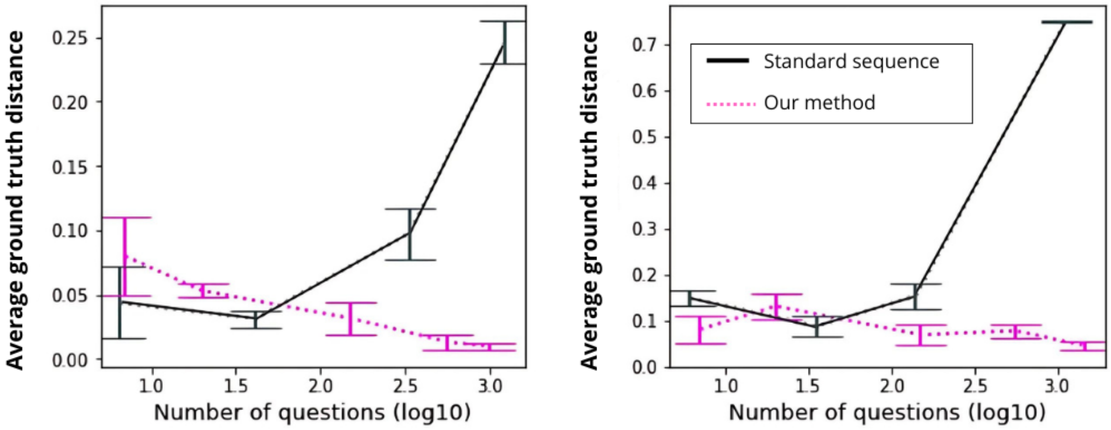


Figure 2.8: Average ground truth distance w.r.t. the number of asked queries for $\epsilon_{max} = 0.05$ (left) and $\epsilon_{max} = 0.1$ (right).

the training set in folds and training the model as many times as the number of folds, each time reserving a different fold for evaluating the model (validation fold). Then, the one-standard-error rule consists in selecting λ as the highest value yielding an average test error on the validation folds lower than the minimum average test error over all λ plus the standard error associated with this minimum. This rule allows selecting the highest amount of regularization (allowing for simpler models) among those that yield minimal average test error. Here the number of folds is set to 3. Also, a grid search is performed over the second hyper-parameter λ_0 whenever it is needed (i.e., for adaptive ℓ_1-penalty and elastic net penalty defined in Remark 2.6).

3.3.1 Recovering the ϵ-min Model

We generate noisy preference data according to the ϵ-min model (see Equation 2.11) with $n = 8$ and different singletons weights, i.e., $h_\epsilon(z) = 1/n \sum_{i=1}^{n} \epsilon_i z_i + (1 - \sum_{i=1}^{n} \epsilon_i) \min_{i \in N}\{z_i\}$ where $(\epsilon_1, \dots, \epsilon_n) = (0.03, 0.03, 0.05, 0.05, 0.02, 0.02, 0.05, 0.05)$. We compare our method based on adaptive ℓ_1-regularization to some baselines, such as the standard ℓ_1-regularization, the unpenalized regression and the use of 2-additivity constraints for an alternative control of model complexity.

In Figure 2.9 we illustrate the one-standard-error rule used to select the optimal value λ^* of the regularization hyper-parameter λ. On the left of Figure 2.9 we show the average test error over the test folds for different values of λ, and λ^* is highlighted (blue star). One can observe (Figure 2.9 on the right) that the number of non-null coefficients decreases as λ increases, and λ^* corresponds to the optimal tradeoff between compactness and generalizing performance. Note that here the second hyper-parameter λ_0 is set to $\lambda_0 = 0.05$.

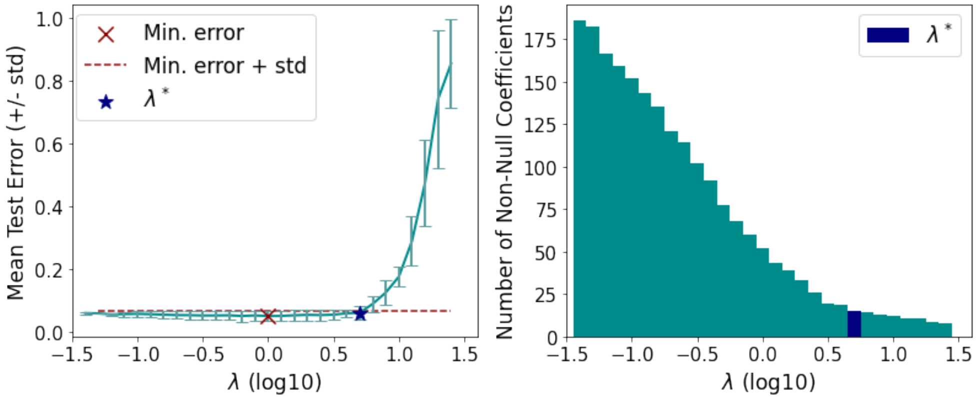


Figure 2.9: Mean test error on the test folds (left) and ℓ_0-norm of models (right) w.r.t. λ.

In Figure 2.10 we show the learned Möbius masses (dashed red) for the adaptive ℓ_1-penalty with λ^* (top left), the ℓ_1-penalty also with optimal regularization parameter λ (top right), the unpenalized regression (bottom left) and the use of 2-additivity constraints (bottom right). For each method, the learned model is superposed to the ground truth model (ϵ-min model represented in gray). It is clear that the regression without any penalty term fails to recover the ϵ-min model; it does not find any compact representation either. It achieves, however, a reasonable generalizing performance on the test set (test error of 0.066). The 2-additive model, while being compact, is far from the ground truth and does not capture interactions involving a large number of attributes, leading to a poor generalizing performance (test error of 0.535). Our approach combines both advantages of the baselines: compactness and optimal generalizing performance (test error of 0.039). In fact, one can see that the ground truth model is exactly recovered. This is not the case with the standard ℓ_1-penalty that includes other coefficients than the non-null ground truth coefficients in the estimated model. This directly illustrates the impact of the violation of Condition **??** for the ϵ-min model, as demonstrated in Example 2.6 for $n = 3$.

3.3.2 Benefit of Adaptive ℓ_1-regularization : Another Illustrative Example

In this section, we provide a second illustration of the benefit of adaptive ℓ_1-regularization compared to standard ℓ_1-regularization in terms of viewpoint interaction selection. To this end, we consider a model ($n = 6$) including 5 interaction terms attached to overlapping groups of viewpoints. The model is given by the following Möbius masses vector: $m_w(\{i\}) = \frac{\epsilon}{n}$ for any $i \in N$, $m_w(S) = \frac{1-\epsilon}{5}$ for any $S \in \{\{1,2\},\{1,2,3\},\{1,2,3,5,6\},\{1,3,4,5,6\},\{1,2,3,4,5,6\}\}$ and $m_w(S) = 0$ everywhere else, with $\epsilon = 0.2$.

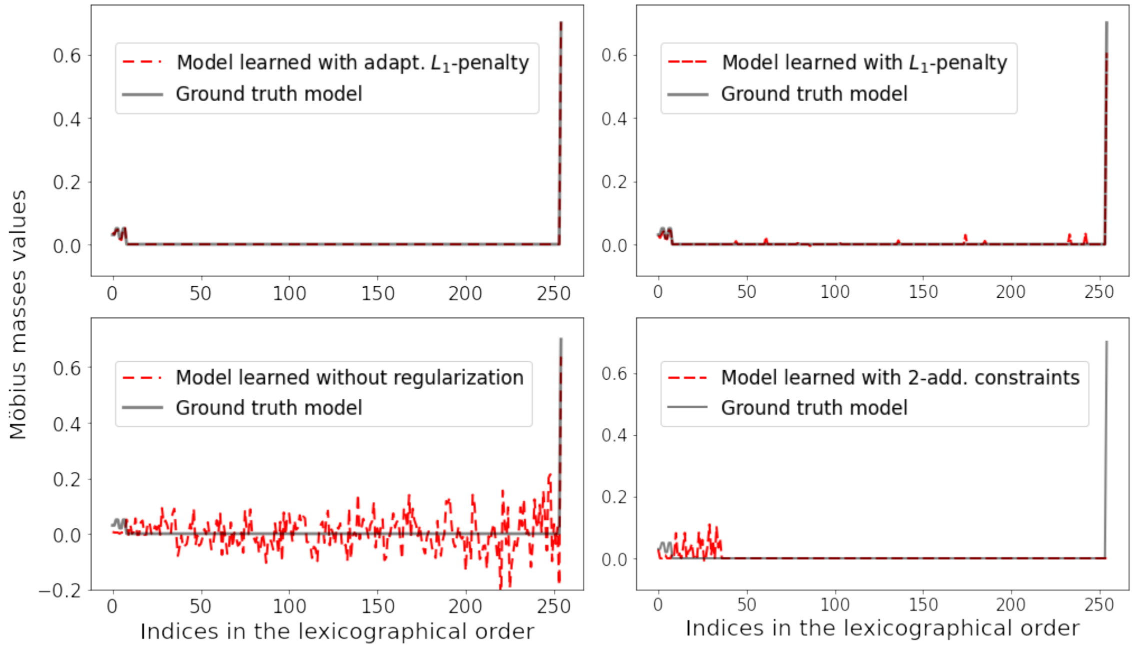


Figure 2.10: Learned models given in the lexicographical order and ground truth model.

We observe the effect of the increase of the hyper-parameter λ for both standard and adaptive ℓ_1-penalization. In Figure 2.11 and Figure 2.12 we represent the *regularization paths* i.e., the evolution of the learned Möbius masses w.r.t λ, for both methods (for the adaptive penalty we take $\lambda_0 = 1$). The non-null coefficients of the ground truth model are highlighted with blue star markers while the null coefficients are displayed with black plain lines. At first glance, the standard ℓ_1-penalization does not succeed to efficiently distinguish ground truth non-null coefficients from null coefficients while the adaptive penalization provides a clear distinction for $\lambda \approx 10^{-0.25}$. Note that for high values of λ, the Möbius masses of singletons remain non-null for both methods. This is quite normal since they are not included in the penalization term (the aim of regularization being only to avoid unecessary non-linearities in the model).

In order to further evaluate and compare the quality of viewpoint interaction selection in both methods we compute the *false discovery rate* (FDR), i.e., the proportion of selected coefficients that are not actually in the ground truth model. We also compute the *false exclusion rate* (FER) which is the proportion of not selected coefficients that are actually in the ground truth model. Figure 2.13 shows the results for standard (left) and adaptive (right) ℓ_1-regularization according to λ. Contrarily to standard ℓ_1-regularization, adaptive ℓ_1-penalty reaches 0% of FDR (gray plain line) and 0% of FER (dashed red line) for $\lambda \in [10^{-1.35}, 10^{-1.1}]$. Thus adaptive ℓ_1-penalty exactly recovers the set of non-null ground truth coefficients. Standard ℓ_1-regularization appears to be less effective since the reduction of the false discovery rate comes at the expense of its false

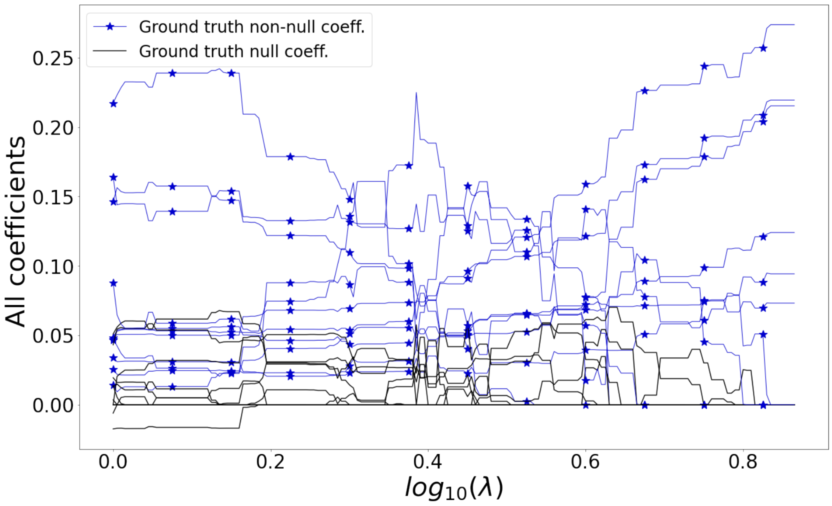


Figure 2.11: Regularization path for standard ℓ_1-penalty.

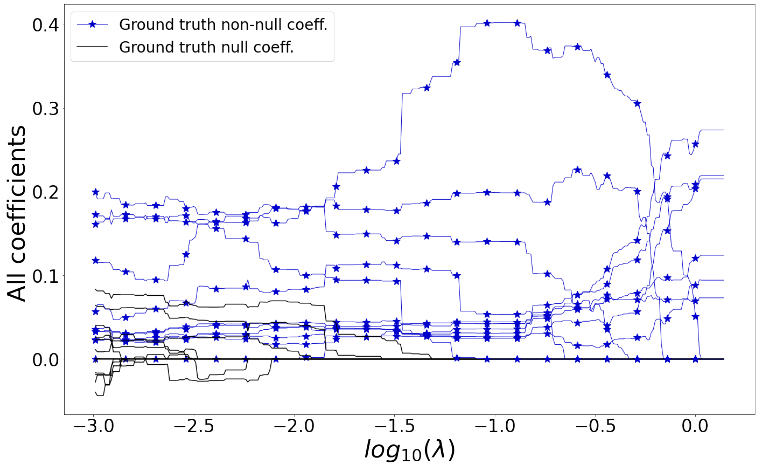


Figure 2.12: Regularization path for adaptive ℓ_1-penalty ($\lambda_0 = 1$).

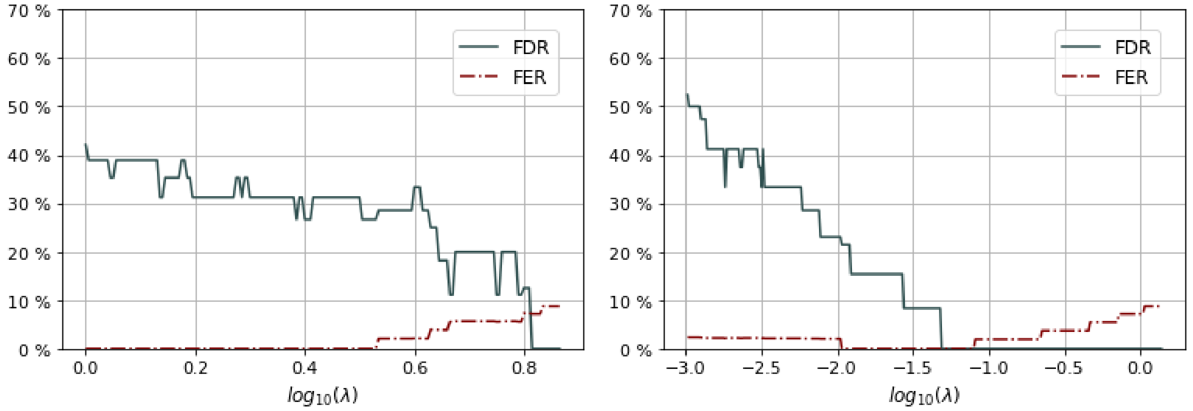


Figure 2.13: FDR and FER for standard (left) and adaptive (right) ℓ_1-penalty w.r.t. λ.

exclusion rate as shown in Figure 2.13 (left).

Stability study. In the previous tests, we assessed the ability of adaptive ℓ_1-regularization to efficiently recover a ground truth model. Now, with another illustrative example, we study the stability of the learned models w.r.t the variability of the training preference data. We use a 5-dimensional CIU model with sparse Möbius transform and generate training sets of preference examples with an increasing level of noise σ. In Figure 2.14 are presented in boxplots the learned Möbius masses with adaptive ℓ_1-regularization obtained for 10 random generations of preference data. From top to bottom are represented the results for increasing values of noise level $\sigma \in \{0, 0.03, 0.05, 0.1\}$. The ground truth model is highlighted with grey bars. For $\sigma = 0$ (top), the exact ground truth model is always recovered over the 10 simulations. Then, increasing the level of noise induces some variability in the learned models. However, for $\sigma = 0.03$ (second from top), very few coefficients that are not in the ground truth model are included in the learned model and the ground truth coefficients are recovered with a nearly constant amplitude. Finally, when the level of noise is high, i.e., $\sigma = 0.1$ (bottom), spurious coefficients such as the grand coalition are included in the learned model and the Möbius masses values are highly variable.

3.3.3 Comparative Performance on Arbitrary Sparse Models

We observed in Section 3.3.1 that CIU used with a 2-additive capacity can fail to properly approximate preference data when the underlying preferences contains higher-order interactions. Also, in Section 3.3.2, we observed that a more sophisticated ℓ_1-regularization is sometimes needed to proceed to a good model selection. In this section, we provide broader tests on synthetic preference data and extend our comparisons to the use of k-additive models for $k = 1, \dots, n-1$, and for an optimal k^* (chosen by cross-

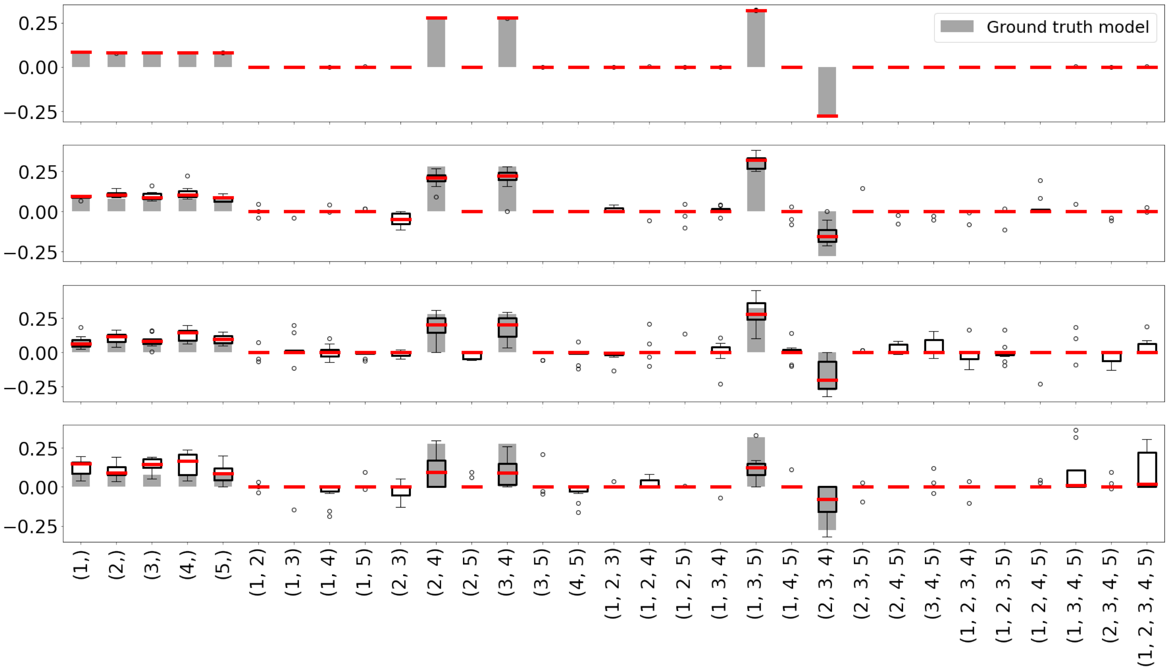


Figure 2.14: Learned Möbius masses for an increasing noise level from top to bottom.

validation). Also, we compare the adaptive ℓ_1-penalty to different penalizations such as the standard ℓ_1-penalty and the elastic net (see Remark 2.6). We finally compare the results of our method to the unpenalized regression method.

First, we generate 20 random CIU models and preference datasets for $n = 8$ (with around 10 non-null coefficients in average). The test error of our approach (referred to as ADA-L1) on test sets is averaged and displayed in Table 2.3 along with the average sparsity of the learned models (ℓ_0-norm of the vector m_w). The quality of the ground truth model retrieval is further assessed with the average gap to the ground truth model computed with the euclidean distance (i.e., $\|\hat{m}_w - m_w^*\|_2^2$) and the false discovery rate (FDR) and false exclusion rate (FER). We also present the results for the baseline methods: the standard ℓ_1-penalty (L1), the elastic net penalty (E-Net), the unpenalized regression (No reg.) and methods that use k-additivity constraints for $k = 2, 4$ and k^* (obtained by cross-validation). Our approach (ADA-L1) clearly outperforms all the methods in terms of compactness, distance to the ground truth model and false discovery rate. Concerning generalizing performance, ADA-L1 outperforms the methods based on k-additive models, especially for $k = 2$ which performs very poorly. The other regularization methods (E-Net and L1) maintain competitive generalizing performance but incorporate non-null ground truth coefficients in the model as the higher falser discovery rate and ℓ_0-norm suggest it. Note that, while having a generalizing performance close to the optimum, the unpenalized regression (No Reg.) provides a dense model and thus is unable to recover an underlying sparse model. As a consequence this method yields a null false exclusion

	Test error	ℓ_0-norm	$\|\hat{m}_w - m_w^*\|_2^2$	FDR	FER
ADA-L1	$\mathbf{0.07 \pm 0.02}$	$\mathbf{16.1 \pm 8.6}$	$\mathbf{0.05 \pm 0.06}$	$\mathbf{0.74 \pm 0.09}$	0.01 ± 0.01
L1	$\mathbf{0.07 \pm 0.01}$	25.1 ± 10.9	0.07 ± 0.10	0.82 ± 0.06	0.01 ± 0.01
E-Net	$\mathbf{0.07 \pm 0.02}$	27.3 ± 9.9	0.08 ± 0.12	0.83 ± 0.07	0.01 ± 0.01
No Reg.	0.09 ± 0.03	206.7 ± 27.1	1.57 ± 2.48	0.97 ± 0.02	$\mathbf{0.00 \pm 0.01}$
2-ADD	0.37 ± 0.18	21.9 ± 2.7	0.61 ± 0.49	0.95 ± 0.05	0.02 ± 0.01
4-ADD	0.13 ± 0.05	148.2 ± 4.8	0.45 ± 0.46	0.98 ± 0.02	0.02 ± 0.01
k^*-ADD	0.09 ± 0.03	147.0 ± 54.4	0.23 ± 0.24	0.97 ± 0.03	0.02 ± 0.02

Table 2.3: Evaluation of the learned CIU models on 20 simulations of synthetic data.

rate.

In Figure 2.15 we show the evaluations obtained for each method using both the generalizing performance (test error) and the number of non-null Möbius masses (ℓ_0-norm). Each curve represents various possible tradeoffs between the test error and the ℓ_0-norm obtained for different values of the regularization hyperparameter λ (for ADA-L1, L1, E-Net) or for different values of k (for k-ADD). For the methods ADA-L1 and E-Net, λ_0 has been priorely set to its best value. We observe that our approach with adaptive ℓ_1-penalty provides significantly better compromises than all the other methods. Moreover, k-additive models perform very poorly, providing models with high ℓ_0-norm and high test error.

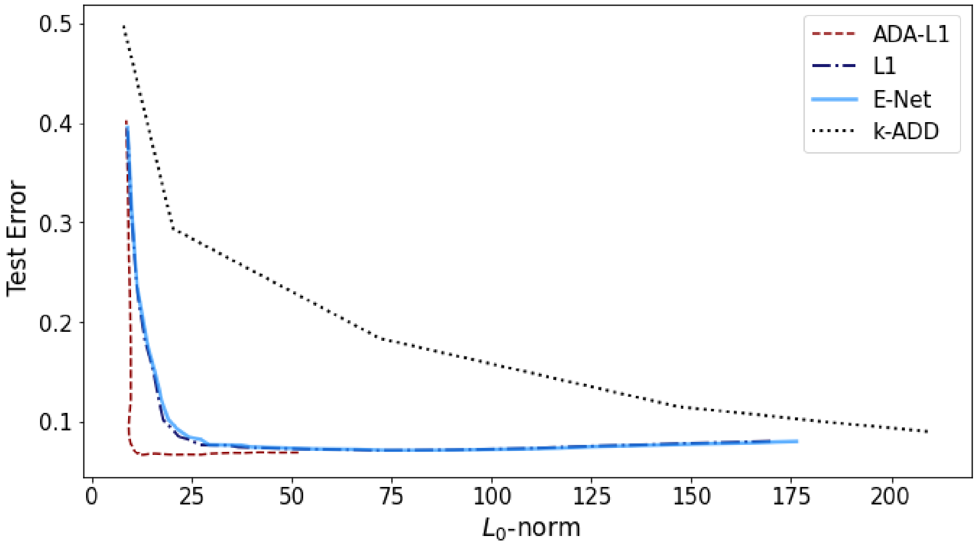


Figure 2.15: Tradeoff test error/ℓ_0-norm depending on the method's hyperparameter.

Finally, we conducted the same experiment with random bi-CIU models and the results are presented in Table 2.4. The results for the learning of both capacities m_w

	Test error	ℓ_0-norm	$\|\hat{m}_w - m_w^*\|_2^2$	FDR	FER
ADA-L1	$\mathbf{0.06 \pm 0.01}$	$\mathbf{16.4 \pm 12.9}$	2.06 ± 0.82	$\mathbf{0.73 \pm 0.11}$	$\mathbf{0.02 \pm 0.01}$
L1	0.07 ± 0.02	25.8 ± 15.8	2.07 ± 0.81	0.84 ± 0.06	$\mathbf{0.02 \pm 0.01}$
E-Net	0.07 ± 0.01	35.8 ± 16.7	$\mathbf{1.15 \pm 0.86}$	0.85 ± 0.04	$\mathbf{0.02 \pm 0.01}$
No Reg.	0.09 ± 0.02	217.1 ± 36.3	58.74 ± 117.65	0.98 ± 0.02	$\mathbf{0.02 \pm 0.01}$
2-ADD	0.30 ± 0.16	20.6 ± 3.8	2.55 ± 0.78	0.95 ± 0.06	$\mathbf{0.02 \pm 0.01}$
4-ADD	0.12 ± 0.05	243.9 ± 72.1	6.94 ± 5.97	0.98 ± 0.02	$\mathbf{0.02 \pm 0.02}$
k^*-ADD	0.09 ± 0.04	215.8 ± 141.5	13.03 ± 14.62	0.94 ± 0.07	$\mathbf{0.02 \pm 0.01}$

Table 2.4: Evaluation of the learned bi-CIU models on 20 simulations of synthetic data.

and $m_{w'}$ are averaged producing a unique result. Here again ADA-L1 produces significantly better results than the other methods in terms of test error, compactness and false discovery rate. Concerning distance to the ground truth, the elastic net penalty provides slightly better results. Remark that all methods perform equally in terms of false exclusion rate.

3.4 Real Data

In this subsection, we test our method for learning sparse Möbius capacity representations on real preference datasets. For this, we use standard monotonic multicriteria decision-making datasets containing overall evaluations of alternatives described by continuous or discrete criteria. Using these datasets, we make the assumption that the learning examples are directly expressed in terms of marginal values.

We use the dataset *Employee Selection* (ESL) from the Weka repository [2], the datasets CPU[3] and Car MPG[4] (MPG) from the UCI repository and the *Movehub city ranking*[5](CITY) dataset. Below, we briefly describe the four datasets:

- ESL: psychologists evaluations on $n = 4$ criteria of some candidates (488) and overall suitability to a position.
- CITY: overall evaluations of quality of life in some cities (216) and $n = 5$ associated descriptors, e.g., purchase power, quality and access to health care.
- CPU: relative performance of some CPUs (209) and $n = 6$ associated technical characteristics, e.g., machine cycle time in nanoseconds, cache memory in kilobytes.

[2] https://www.openml.org
[3] https://archive.ics.uci.edu/dataset/29/computer+hardware
[4] https://archive.ics.uci.edu/dataset/9/auto+mpg
[5] https://www.kaggle.com/datasets/blitzr/movehub-city-rankings

	ESL	CITY	CPU	MPG
ADA-L1	**5.42 ± 2.38**	6.14 ± 3.82	**6.11 ± 1.83**	7.69 ± 2.48
L1	5.73 ± 2.81	6.69 ± 4.29	7.81 ± 3.45	**7.58 ± 2.8**
E-Net	5.93 ± 2.93	6.99 ± 5.67	17.44 ± 13.16	23.44 ± 12.51
No Reg.	12.71 ± 1.60	23.26 ± 4.77	42.04 ± 9.15	55.83 ± 14.69
2-ADD	7.80 ± 1.19	9.09 ± 1.61	9.73 ± 1.84	8.21 ± 1.58
4-ADD	12.71 ± 1.60	22.58 ± 4.42	36.73 ± 7.47	36.54 ± 12.29
k^*-ADD	5.77 ± 2.77	**5.97 ± 3.52**	12.05 ± 9.94	12.79 ± 12.12

Table 2.5: Average ℓ_0-norm for ADA-L1 and for the baselines on real datasets.

ESL	CITY	CPU	MPG
0.22 ± 0.04	0.05 ± 0.03	0.12 ± 0.05	0.15 ± 0.07

Table 2.6: Average test error for ADA-L1 on real datasets.

- MPG: city-cycle fuel consumption in miles per gallon of some cars (398) and $n = 7$ associated technical characteristics, e.g., weight, acceleration, model year.

These datasets of overall evaluations are turned into datasets of preference and indifference statements by randomly drawing pairs of alternatives (without replacing them) and comparing their global scores. The criteria associated with a decreasing monotonicity are multiplied by -1 and the marginal value values are made commensurate by means of linear normalization.

We compare ADA-L1 and the baseline methods in terms of test error and number of non-null coefficients of the learned models (ℓ_0-norm). The results are averaged over 100 simulations for each dataset. For each simulation, the models are trained on 80% of the dataset and tested over the 20% left with a random split. In Table 2.5 are presented the average ℓ_0-norm of the learned models for the different methods. The results leading to the smaller ℓ_0-norms are highlighted in bold and the second-best results are underlined. ADA-L1 provides very sparse models with significantly lower ℓ_0-norms than the one obtained with the baseline methods. This model compacity is obtained at no cost in terms of generalizing performance since ADA-L1 provides test errors similar to the baseline methods. We indeed performed pairwise t-tests to test the significance of the difference in test error between all the methods and we obtained p-values of magnitude 0.5. The test error numerical values obtained for ADA-L1 are provided in Table 2.6. This suggests that ADA-L1 is able to identify the few criteria coalitions that really matter in the preference value system underlying each dataset.

4 Conclusion

In this chapter, we have introduced a new approach to learn both marginal values and capacities in CIU and bi-CIU models in the context of decision-making under uncertainty (DMU) and multi-criteria/attribute decision-making (MCDM/MADM). We first proposed a variant of the tradeoff method for both DMU and MADM contexts to learn marginal utility functions which appears to be more robust than usual elicitation methods based on standard sequences. Then we presented a method to learn sparse Möbius representations of capacities using adaptive ℓ_1-regularization. It determines where are the Möbius masses that really matter to define the capacity. This reveals those interacting subsets of criteria that must be kept in the general Choquet integral model to fit the observed preferences. One important advantage of this approach is that interacting subsets of any size can be included in the model. No prior restriction on the size of interaction factors is made, they are derived from the database of preference examples.

An important aspect concerns the complexity of the learning task. The linear reformulation of problem $\mathcal{P}'$ introduced in Section 2 includes $2^{n+1} + 2|I| + |P|$ variables and $\sum_{k=1}^{n} k\binom{n}{k} + |I| + |P| + 1$ constraints. Therefore the problem to be solved grows exponentially with the number of viewpoints n. It remains tractable up to a dozen of viewpoints which covers most of practical cases in MCDM. In order to improve scalability of the method, several options could be investigated. First, a hierarchical structure over criteria can be used which may drastically reduce the number of criteria to be aggregated at every level and therefore the size of the learning problem. This idea was implemented in [Bresson et al., 2021] to learn 2-additive capacities and could be extended to learn general capacities [Bresson, 2022]. Another option would be to leverage the dual formulation of the optimization problem $\mathcal{P}'$ as in *kernel-based* machine learning methods. This option is investigated in the next chapter.

Beside scalability, several natural extensions of this work could be considered. First, the construction of compact representations of CIU is based on Equation 2.12 that combines Möbius masses and terms of type $\min_{i\in S}\{u_i(x_i)\}$. Alternative representations exist for CIU and bi-CIU, combining Möbius masses and factors of type $\max_{i\in S}\{u_i(x_i)\}$. They could lead to compact representations as well. This suggests extending our approach and combining min and max factors to produce even more compact representations of capacities. Another extension could be to adapt our approach to other decision models allowing interacting criteria. For example, the multilinear marginal value model [Keeney et al., 1993] admits a representation in terms of Möbius masses similar to Equation 2.12 where min factors are substituted by products $\prod_{i\in S} u_i(x_i)$. Clearly, the learning approach we have proposed here for the capacity identification also applies to this model with very minor modifications. Such extensions are investigated along with the scalability question in the next chapter.

Chapter 3
A Unified Approach to Learn Sparse Preference Models with Interactions

Contents

Summary

In this chapter, we consider a large class of preference models that allow for the synthesis of conflicting and potentially *interacting* viewpoints. Allowing viewpoint interactions in a preference model increases the complexity of the preference learning task due to the combinatorial nature of the possible interactions. Here, we propose a general approach to learn a preference model in which the interaction pattern is revealed from preference data and kept as simple as possible. Within a *unified* framework, we consider weighted aggregation functions such as *multilinear utilities* and *Choquet integrals*, which allow representations that include non-linear terms capturing the joint benefit or penalty associated with certain combinations of viewpoints.. The weighting coefficients, known as Mobius masses, model positive or negative synergies among viewpoints. We propose an approach to learn the Mobius masses, based on *iterative reweighted least squares* for sparse recovery, and *dualization* to improve scalability. This approach is applied to learn sparse representations of the multilinear utility and conjunctive/disjunctive forms of the discrete Choquet integral from preferences examples, in aggregation problems possibly involving more than 20 viewpoints. This chapter is based on the following publications: [Herin et al., 2022c, 2023b].

Introduction

In this chapter, we aim to learn preference models that are both simple and explainable, yet flexible enough to accurately model human preferences and decision behaviors. As outlined in Chapter 1 (see, for instance, Examples 1.3 and 1.5), the possible presence of interactions between viewpoints prevents representing preferences using simple linear models such as weighted arithmetic means. Therefore, we consider here more sophisticated weighted evaluation models that include nonlinear terms, which measure the joint benefit or penalty associated with certain groups of viewpoints. Key examples of these models include the *multilinear utility* and the *Choquet integral* (see Definitions 1.15 and 1.9), where interactions are respectively represented by product and minimum or maximum operations.

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 viewpoints, interactions may appear in any of the $2^n - 1 - n$ subsets of viewpoints including more than one element. For $n = 10$ viewpoints 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, using for instance k-additivity constraints (see Definition 1.10). These restrictions are very often used to obtain a prior model complexity reduction ($k = 2$ being the most common choice) [Grabisch et al., 2008, Grabisch and Labreuche, 2010, Tehrani and Hüllermeier, 2013, Hüllermeier and Tehrani, 2013, Galand and Mayag, 2017b, Ah-Pine et al., 2018, Bresson et al., 2021, Tehrani, 2021, Pelegrina et al., 2020a].

On the other side, as outlined in Chapter 2 (see for instance Example 2.3), these cardinal-based prior restrictions eliminate very simple and natural representations of preferences that require larger interactions. In contrast, using a sparsity-inducing penalty in the learning problem allows useful groups to emerge from preference data, thus providing a model that is as simple as possible and fits the preference examples well. However, the absence of prior restrictions on capacities comes at the expense of computation times and scalability, and the proposed methods are usually implemented on problems involving less than 10 viewpoints [Anderson et al., 2014, Adeyeba et al., 2015, Pinar et al., 2017, de Oliveira et al., 2022, Herin et al., 2022a, 2024c].

Contributions and Chapter Organization Our contribution in this chapter is to propose a *scalable algorithm* to learn *sparse representations of interactions* from preference examples, for the multilinear and Choquet models and any other weighted aggregation function taking the form of a sum of disjunctive or conjunctive interaction terms. To this end, we build on the *iteratively re-weighted least squares* (IRLS) method [Daubechies et al., 2010, Beck, 2015], which consists of approximating an ℓ_1-norm minimization problem through a sequence of least squares problems, typically easier to solve. Specifically, after formulating a general ℓ_1-regularized preference learning problem (Section 1), we first show that this problem can be approximated using an IRLS sequence (Section 2.1), and then, leveraging *Lagrangian duality* in the spirit of *support vector machines* (see Subsection 3.1.2 of Chapter 1), we show that each sub-problem of the sequence admits an efficient dual formulation (Section 2.2). The benefit of this approach is finally illustrated on problems involving up to more than 20 viewpoints using synthetic preference data (Section 3.1). It is also applied to judicial decision-making in divorce cases using real-world data (Section 3.2).

Notations The transpose of a matrix/vector v is denoted by $v^\top$ and $v * u$ denotes the element-wise product. Additionally, notations $\mathbf{0}$ and $\mathbf{1}$ are used to denote vectors of appropriate dimension whose components all equal 0 and 1 respectively. Finally, recall that N denotes the set of viewpoints, i.e., $N = \{1, \ldots, n\}$ and that the notation $S \subseteq N$ excludes the empty set by convention. Also, for any $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ and $S \subseteq N$, x_S refers to the restriction of x to the components $x_i, i \in S$.

1 A General Capacity-based Preference Model

In this chapter, we focus on learning non-linear aggregation functions such as the *multilinear utility* and the *Choquet integral* (see Definitions 1.15 and 1.9), and therefore, we assume that the alternatives in the decision problem are described by their marginal utilities with respect to n viewpoints (elicited beforehand), i.e., by vectors of the form $x = (x_1, \ldots, x_n) \in \mathcal{X} = [0, 1]^n$. Both the multilinear and the Choquet model allow modeling interaction between viewpoints by means of a capacity w (see Definition 1.8) that attaches a weight to any possible group of interacting viewpoints. While the two aggregation functions, denoted respectively by ML_w and C_w, may not seem to function the same way at first glance, it is interesting to note that they admit a similar formulation using the Möbius transform m_w (see Definition 1.4) of capacity w. To highlight this, we first recall the definition of ML_w from Möbius masses [Owen, 1975]:

$$\mathrm{ML}_w(x) = \textstyle\sum_{S \subseteq N} m_w(S) \prod_{i \in S} x_i \tag{3.1}$$

Similarly, $C_w(x)$ admits several reformulations from m_w [Chateauneuf and Jaffray, 1989, Grabisch et al., 2009]:

$$C_w(x) = \textstyle\sum_{S \subseteq N} m_w(S) \min_{i \in S}\{x_i\} \quad (\textit{conjunctive form}) \tag{3.2}$$

$$C_w(x) = \textstyle\sum_{S \subseteq N} m_{\bar{w}}(S) \max_{i \in S}\{x_i\} \quad (\textit{disjunctive form}) \tag{3.3}$$

where $\bar{w}$ is the *dual* of w, i.e., the capacity defined by $\bar{w}(S) = w(N) - w(N \setminus S), \quad S \subseteq N$.

These formulations suggest that both ML_w and C_w might admit a compact representation when the Möbius inverse is *sparse* (i.e., when the vector of Möbius masses includes many zeros or small values that will not significantly impact the calculation). This is illustrated in the following toy example with $n = 3$ viewpoints.

Example 3.1. *Let $N = \{1, 2, 3\}$ and $w, \bar{w}$ defined on N by:*

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
$w(S)$	0.1	0.2	0.3	0.3	0.4	0.5	1.0
$m_w(S)$	0.1	0.2	0.3	0.0	0.0	0.0	0.4
$\bar{w}(S)$	0.5	0.6	0.7	0.7	0.8	0.9	1.0
$m_{\bar{w}}(S)$	0.5	0.6	0.7	-0.4	-0.4	-0.4	0.4

While w is dense (i.e., $w(S) > 0,$ for any $S \subseteq N$), its Möbius transform m_w is sparse since the Möbius masses of the interaction subsets (of size strictly higher than 1) all equal zero except the grand coalition mass $m_w(\{1, 2, 3\})$. Therefore ML_w and C_w admit a simple formulation using respectively Equations 3.1 and 3.2:

$$\mathrm{ML}_w(x) = 0.1\ x_1 + 0.2\ x_2 + 0.3\ x_3 + 0.4\ x_1x_2x_3 \tag{3.4}$$

$$C_w(x) = 0.1\ x_1 + 0.2\ x_2 + 0.3\ x_3 + 0.4 \min\{x_1, x_2, x_3\} \tag{3.5}$$

Here the disjunctive form of C_w (Eq. 3.3) is less interesting because $m_{\bar{w}}$ is less sparse than m_w. However, $C_{\bar{w}}$ can be simply described using the disjunctive form, as the dual capacity of $\bar{w}$ is w. Then, using m_w and Equation 3.3, we have:

$$C_{\bar{w}}(x) = 0.1x_1 + 0.2x_2 + 0.3x_3 + 0.4 \max\{x_1, x_2, x_3\} \tag{3.6}$$

In order to factorize and generalize Equations 3.1-3.3, we will now introduce a unifying framework based on a general *capacity-based preference model*, including C_w and ML_w as special cases, that associates to any alternative $x \in \mathcal{X}$, the value:

$$F_m(x) = \textstyle\sum_{S \subseteq N} m_S \phi_S(x_S) \tag{3.7}$$

where for any $S \subseteq N$, m_S is the Möbius mass on S and ϕ_S aggregates the quantities $x_i, i \in S$ to define the *interaction term* $\phi_S(x_S)$. Thus ϕ_S is the product if F_m is the multilinear utility and ϕ_S is the min (resp. max) operation if F_m is the conjunctive (resp. disjunctive) form of the Choquet integral. Note that function $F_m(x)$ reads as the following inner product $F_m(x) = m^\top \phi(x)$ where $m = (m_S)_{S \subseteq N}$ and $\phi : \mathbb{R}^n \to \mathbb{R}^{2^n-1}$ is a nonlinear mapping function that maps x to $\phi(x) = (\phi_S(x_S))_{S \subseteq N}$. Both vectors m and $\phi(x)$ are indexed by the subsets $S \subseteq N$ numbered in lexicographic order.

Remark 3.1 (a general interaction function). Interaction function ϕ_S could possibly be other nonlinear factors than the product, min or max, as long as the aggregation function F_m remains non-decreasing, to ensure the monotonicity of the preference model (see Definition 1.6). Conditions on ϕ_S for F_m to be non-decreasing are provided in [Kolesarova et al., 2012] (for m corresponding to the Möbius transform of a capacity monotonic w.r.t. set inclusion).

Our objective is now to learn a sparse representation of m based on a training set of preference statements $\{(x^\ell, x'^\ell)\}_{\ell \in P}$ and possibly of indifference statements $\{(x^\ell, x'^\ell)\}_{\ell \in I}$ where for any $\ell \in P$, $x^\ell \succsim x'^\ell$ and for any $\ell \in I$, $x^\ell \sim x'^\ell$. The learning problem naturally formulates as a regularized empirical risk minimization (RERM) problem (see Definition 1.21) where we minimize both the error on the preference examples (using the *pref-hinge loss*; see Definition 1.28) and the ℓ_1-norm of the Möbius vector. The learning problem is thus formulated as follows:

$$(\mathcal{P}) \min_{m \in \mathbb{R}^{2^n-1},\ m^\top \mathbf{1}=1} \sum_{\ell \in P} (\delta - (m^\top \phi(x^\ell) - m^\top \phi(x'^\ell)))_+ + \sum_{\ell \in I} |m^\top \phi(x^\ell) - m^\top \phi(x'^\ell)| + \lambda \|m\|_1$$

The hyper-parameter $\lambda \geq 0$ controls the level of regularization and δ is a positive discrimination threshold used to separate preference from indifference situations.

From now on, for any pair of alternatives (x^ℓ, x'^ℓ), the difference $\phi(x^\ell) - \phi(x'^\ell)$ is denoted by Δ^ℓ. Then, similarly as in Chapter 2 (see Section 2.2.3), Problem $\mathcal{P}$ can be solved by linear programming using the following linearization:

$$\begin{aligned}
(\mathcal{P}) \qquad & \min \sum_{\ell \in P} \epsilon_\ell + \sum_{\ell \in I} (\epsilon_\ell^- + \epsilon_\ell^+) + \lambda \sum_{j=1}^{2^n-1} (a_j + b_j) \qquad (3.8)\\
& m_j = a_j - b_j, \quad j = n+1, \ldots, 2^n - 1\\
& m^\top \Delta^\ell + \epsilon_\ell \geq \delta, \ \ell \in P\\
& m^\top \Delta^\ell + \epsilon_\ell^+ - \epsilon_\ell^- = 0, \ \ell \in I\\
& m^\top \mathbf{1} = 1, \ \epsilon_\ell \geq 0, \ \ell \in P, \ \ \epsilon_\ell^+, \epsilon_\ell^- \geq 0, \ \ell \in I, \ \ a_j, b_j \geq 0, \ \ j = 1, \ldots, 2^n - 1
\end{aligned}$$

where variable m_j is the j^{th} component of vector m, and variables (a_j, b_j), ϵ_ℓ, and $(\epsilon_\ell^+, \epsilon_\ell^-)$ are respectively used for the linearization of $|m_j|$, the error made on the preference example $x^\ell \succ x'^\ell$, and on the indifference $x^\ell \sim x'^\ell$.

Despite a simple linearization, the obtained linear program still drags a number of variables exponential in n (i.e., $2(2^n - 1) + |P| + 2|I|$) and thus is hardly solvable for more than a dozen of viewpoints using standard linear programming numerical solvers. A way to bypass this issue could be to reduce the size of the learning optimization problem by resorting to k-additivity constraints, i.e., enforcing $m_S = 0$, for any $S \subseteq N$ such that $|S| > k$. However, as discussed in Chapter 2 (see Example 2.3), such a reduction significantly limits the ability of the preference model F_m to capture natural preferences. This is also illustrated in Example 3.1, where the models given by by Equations 3.4-3.6, while admitting sparse Möbius transforms and simple formulations, are n-additive and could not be well approximated by k-additive models. Thus, we propose not to resort to k-additivity constraints and solve Problem $\mathcal{P}$ with the full set of Möbius variables. Then, to handle problems involving more than a dozen of viewpoints, we propose a more scalable optimization method than solving $\mathcal{P}$ using linear programming. The approach is formulated for the multilinear and Choquet models and any other instance of model F_m (see Equation 6.1), and relies on *iteratively re-weighted least squares* and *dualization* as explained in the next section.

2 A Dual Iterative re-Weighted Least Squares (IRLS) Algorithm

For the sake of scalability, we propose to solve $\mathcal{P}$ by solving a sequence of subproblems that admit an efficient dual formulation. More precisely, we use an *iteratively reweighted least square* (IRLS) algorithm that consists in approximating the solution of a ℓ_1-norm minimization problem with a sequence of least squares problems. In the following, we first present the underlying idea and theoretical foundations of IRLS sequences, before deriving an IRLS sequence for Problem $\mathcal{P}$. Then, after introducing some background on *lagrangian duality*, illustrated with *support vector machines*, we show that the subproblems of the proposed IRLS sequence admit compact dual formulations.

2.1 IRLS for Sparse Preference Learning

2.1.1 Variational Formulation of the ℓ_1-norm

ℓ_1-norm optimization can be linked to least squares problems through a *quadratic variational formulation* of the ℓ_1-norm, which allows the absolute value function to be

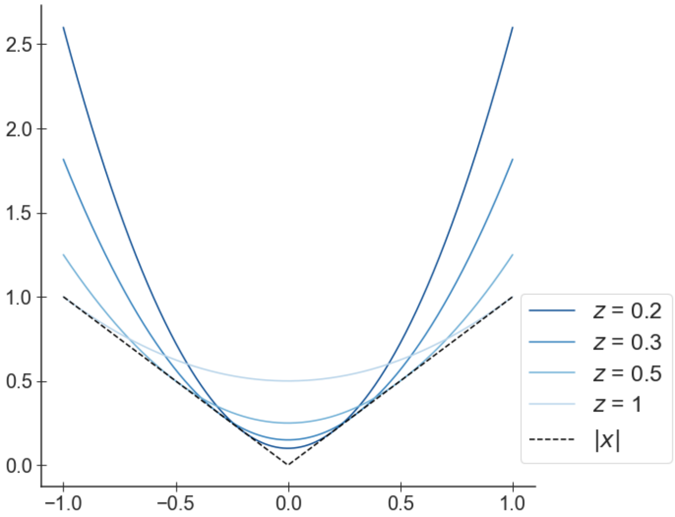


Figure 3.1: Functions $g_z : x \mapsto \frac{1}{2}(\frac{x^2}{z} + z)$ for different z values and function $g : x \mapsto |x|$.

expressed as the infimum of quadratic functions [Black and Rangarajan, 1996, Rocha et al., 2009, Bach et al., 2012], i.e., for any $x \in \mathbb{R}$:

$$|x| = \frac{1}{2} \min_{z \geq 0} \frac{x^2}{z} + z \tag{3.9}$$

For any $x \neq 0$, Equation 3.9 can be simply derived by observing that for any $z > 0$, using the arithmetic-geometric mean inequality i.e., $\frac{a+b}{2} \geq \sqrt{ab}$, for $a = \frac{x^2}{z}$ and $b = z$, we have $|x| \leq \frac{1}{2}(\frac{x^2}{z} + z)$. Finally, we recover Equation 3.9 by remarking that the lower bound $|x|$ is reached for $z = |x| > 0$. The equality also holds for $x = 0$ using the convention $\frac{0}{0} = 0$. It is illustrated in Figure 3.1, where we can observe that function $g : x \mapsto |x|$ (dashed black line) is the infimum of the quadratic functions $g_z : x \mapsto \frac{1}{2}(\frac{x^2}{z} + z)$, $z \in \mathbb{R}_+^*$, which are shown in blue for $z = \{0.2, 0.3, 0.5, 1\}$.

Using Equation 3.9, ℓ_1-regularized RERM problems of the form $\min_m R_{emp}(m) + \lambda \|m\|_1$, where m is a model parameter vector of size d and R_{emp} is the empirical risk on some dataset (see Definition 1.20), can be reformulated as a two-block optimization problem, involving a d-dimensional vector $z = (z_1, \dots, z_d)$ of auxiliary variables:

$$\min_{m, z \geq \mathbf{0}} R_{emp}(m) + \frac{\lambda}{2} \sum_{j=1}^{d} (\frac{m_j^2}{z_j} + z_j) \tag{3.10}$$

Then, the *iterative re-weighted least squares* (IRLS) algorithm [Grandvalet, 1998, Daubechies et al., 2010, Bach et al., 2012, Beck, 2015] consists in *alternatively minimizating* w.r.t. m and z. More precisely, if $m^{(0)}$ denotes an initial model parameter vector, and $z^{(0)} = |m^{(0)}|$ (component-wise), at each iteration $k \geq 0$, the two following optimization tasks are successively performed:

$$m^{(k+1)} \in \arg\min_{m} R_{emp}(m) + \frac{\lambda}{2}\sum_{j=1}^{d}\frac{m_j^2}{z_j^{(k)}} \tag{3.11}$$

$$z^{(k+1)} \in \arg\min_{z\geq\mathbf{0}} \frac{\lambda}{2}\sum_{j=1}^{d}(\frac{(m_j^{(k+1)})^2}{z_j} + z_j) \tag{3.12}$$

By Equation 3.9 the solution of Problem 3.12 is $z_j^{(k+1)} = |m_j^{(k+1)}|, \quad j = 1, \ldots, d$. Thus, the algorithm given above reduces to a sequence of RERM problems with a weighted (squared) ℓ_2-regularization, where the weights are iteratively updated, i.e., :

$$m^{(k+1)} \in \arg\min_{m} R_{emp}(m) + \frac{\lambda}{2}\sum_{j=1}^{d}\frac{m_j^2}{|m_j^{(k)}|}$$

Intuitively, this iterative procedure allows sparsity to be recovered by increasingly penalizing non-significant coefficients (which end up vanishing). The interest of such procedure lies in the fact that least squares problems are typically easy to solve.

For instance, let us consider the learning of a linear model $h(x) = m^\top x$ using regression examples $(x^\ell, y^\ell) \in \mathbb{R}^d \times \mathbb{R}$ stored in matrices $\boldsymbol{X} = (x^\ell)_\ell$ and $\boldsymbol{Y} = (y^\ell)_\ell$. If we use the square loss, i.e., $R_{emp}(m) = \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{X}m\|_2^2$, the problem given by Equation 2.1.1 can be written as: $m^{(k+1)} \in \arg\min_m \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{X}m\|_2^2 + \frac{\lambda}{2}m^T\boldsymbol{D}_k^{-1}m$, where $\boldsymbol{D}_k^{-1}$ is a diagonal matrix whose diagonal contains the weights $|m_j^{(k)}|, j = 1, \ldots, d$. Then, it can easily be checked that each subproblem admits the closed-form solution $m^{(k+1)} = (\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{D}_k^{-1})^{-1}\boldsymbol{X}^\top\boldsymbol{Y}$. For further illustrative examples, the reader may refer to the blog [Bach, 2019].

Here, we will show that the IRLS method allows the approximation of the solution of Problem $\mathcal{P}$ using a sequence of ℓ_2-regularized problems admitting a compact dual form whose size is no longer exponential in n the number of viewpoints, but linear in $|P| + |I|$, the number of preference and indifference examples. Before that, we give general results from [Beck, 2015] on the convergence of the alternating minimization (AM) algorithm, allowing to guarantee the convergence of the proposed IRLS sequence.

2.1.2 Convergence of the Alternating Minimization Algorithm

Below, we give a general convergence result for the AM algorithm due to [Beck, 2015] for two-block optimization problems of the form:

$$\min_{m\in\mathbb{R}^{n_1}, z\in\mathbb{R}^{n_2}} H(m, z) := f(m, z) + g_1(m) + g_2(z) \tag{3.13}$$

where $n_1, n_2 \in \mathbb{N}^*$ and f, g_2, g_1 satisfy the following assumptions:

(i) functions $g_1 : \mathbb{R}^{n_1} \to \mathbb{R}$ and $g_2 : \mathbb{R}^{n_2} \to \mathbb{R}$ are closed and proper convex functions assumed to be subdifferentiable over their domains $\operatorname{dom} g_1$ and $\operatorname{dom} g_2$ (see Definitions 1.24,1.26 and 1.27).

(ii) function f is a continuously differentiable convex function over $\operatorname{dom} g_1 \times \operatorname{dom} g_2$.

(iii) the gradient of f w.r.t. the first block of variables m, denoted by $\nabla_1 f$, is uniformly Lipschitz continuous over dom g_1 with constant $0 < L_1 < \infty$, i.e., for any $m \in \operatorname{dom} g_1, z \in \operatorname{dom} g_2$ and $d_1 \in \mathbb{R}^{n_1}$ such that $m + d_1 \in \operatorname{dom} g_1$, we have:

$$\|\nabla_1 f(m + d_1, z) - \nabla_1 f(m, z)\|_2 \leq L_1 \|d_1\|_2 \tag{3.14}$$

Denote by $\nabla_2 f$ the gradient of f w.r.t. the second block of variables z, and by L_2 its Lipschitz constant. $\nabla_2 f$ may be not uniformly Lipschitz continuous, i.e., $L_2 = +\infty$.

In this setting, the AM algorithm consists of iteratively optimizing over each block of variables as follows:

Algorithm 3.1: Alternating Minimization (AM) Algorithm

```
Inputs: $m^{(0)} \in \operatorname{dom} g_1$
$z^{(0)} \leftarrow \arg\min_z f(m^{(0)}, z) + g_2(z)$
for $k = 0, \dots$ do
    $m^{(k+1)} \in \arg\min_m f(m, z^{(k)}) + g_1(m)$
    $z^{(k+1)} \in \arg\min_z f(m^{(k+1)}, z) + g_2(z)$
```

The following theorem guarantees the convergence of the sequence $H(m^{(k)}, z^{(k)})$ towards the minimum value H^* of Problem 3.13 under Assumptions (i)-(iii).

Theorem 3.1 (Theorem 3.3 in [Beck, 2015]). *Let $\left\{m^{(k)}, z^{(k)}\right\}_{k\geq 0}$ be the sequence generated by the AM algorithm. Then for all $k \geq 2$:*

$$H\left(m^{(k)}, z^{(k)}\right) - H^* \leq \max\left\{\left(\frac{1}{2}\right)^{\frac{k-1}{2}}\left(H\left(m^{(0)}, z^{(0)}\right) - H^*\right), \frac{8\min\{L_1, L_2\} R^2}{k-1}\right\}$$

where H^* is the optimal value of Problem 3.13, X^* the set of optimal solutions and $R = \max_{x\in\mathbb{R}^{n_1\times n_2}} \max_{x^*\in X^*}\left\{\|x - x^*\|_2 : H(x) \leq H\left(m^{(0)}, z^{(0)}\right)\right\}$.

2.1.3 IRLS for Sparse Preference Learning

Using Theorem 3.1, we now establish Proposition 3.1 providing an IRLS algorithm that approximatively solves Problem $\mathcal{P}$. The proof follows a similar line of reasoning to the analysis of IRLS sequences in [Beck, 2015], while taking into account the specificities of Problem $\mathcal{P}$, in particular the non-differentiability of the pref-hinge loss.

Proposition 3.1. *Consider the sequence $m^{(k)}$, initialized with any $m^{(0)}$ such that $\mathbf{1}^\top m^{(0)} = 1$, and defined for any $k \geq 1$ by:*

$$m^{(k+1)} \in \underset{m\in\mathbb{R}^d \text{ s.t. } m^\top\mathbf{1}=1}{\arg\min} \sum_{\ell\in P}(\delta - m^\top\Delta^\ell)_+ + \sum_{\ell\in I}|m^\top\Delta^\ell| + \frac{\lambda}{2}\sum_{j=1}^{d}\frac{m_j^2}{\sqrt{(m_j^{(k)})^2+\eta^2}} \tag{3.15}$$

where $d = 2^n - 1$ and $\eta > 0$ is a smoothing parameter. Let also J denotes the objective function of $\mathcal{P}$ and J^* its optimum. Then, for any $k \geq 2$:

$$J(m^{(k)}) - J^* \leq \max\left\{\left(\frac{1}{2}\right)^{\frac{k-1}{2}}\left(J_\eta(m_{(0)}) - J_\eta^*\right), \frac{16\lambda R^2}{\eta(k-1)}\right\} + \lambda d\eta$$

where J_η is a surrogate of $\mathcal{P}$ in which the ℓ_1-norm is substituted by $\sum_{j=1}^d \sqrt{m_j^2+\eta^2}$ (and J_η^* is its optimum), and R is a constant independent of k.

Proof. *Let $\eta > 0$ be a smoothing parameter and $\mathcal{P}_\eta$ the associated surrogate problem of $\mathcal{P}$ where the absolute value $|m_j|$ is replaced by the differentiable term $\sqrt{m_j^2+\eta^2}$ (which approaches $|m_j|$ when $\eta \to 0$):*

$$(\mathcal{P}_\eta) \quad \min \sum_{\ell\in P}(\delta - m^\top\Delta^\ell)_+ + \sum_{\ell\in I}|m^\top\Delta^\ell| + \lambda\sum_{j=1}^{d}\sqrt{m_j^2+\eta^2}$$
$$\text{s.t. } m^\top\mathbf{1} = 1$$

Remark that Problem $\mathcal{P}_\eta$ can be reformulated in an unconstrained form:

$$(\mathcal{P}_\eta) \quad \min \sum_{\ell\in P}(\delta - m^\top\Delta^\ell)_+ + \sum_{\ell\in I}|m^\top\Delta^\ell| + \lambda\sum_{j=1}^{d}\sqrt{m_j^2+\eta^2} + \mathbb{1}_{\mathcal{M}}(m)$$

with $\mathcal{M} = \{m \in \mathbb{R}^d | m^\top\mathbf{1} = 1\}$, and $\mathbb{1}_{\mathcal{M}}(m) = 0$ if $m \in \mathcal{M}$ and $+\infty$ otherwise. Then, using Equation 3.9 for $x = \sqrt{m_j^2+\eta^2}, j = 1, \ldots, d$ yields:

$$\sum_{j=1}^{d}\sqrt{m_j^2+\eta^2} = \frac{1}{2}\min_{z\geq\frac{\eta}{2}\mathbf{1}}\sum_{j=1}^{d}(\frac{m_j^2+\eta^2}{z_j} + z_j)$$

Note that the optimization is performed over $z \geq \frac{\eta}{2}\mathbf{1}$, as this avoids singularities at 0 *while not affecting the minimum, which is reached at $z_j = \sqrt{m_j^2 + \eta^2} \geq \eta \geq \frac{\eta}{2}$. This leads to reformulate $\mathcal{P}_\eta$ as a problem involving two blocks of variables $(m, z) \in \mathbb{R}^d \times \mathbb{R}^d$:*

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

$$\text{with} \begin{cases} f(m, z) = \frac{\lambda}{2} \sum_{j=1}^{d} (\frac{m_j^2+\eta^2}{z_j} + z_j) \\ g_1(m) \quad = \sum_{\ell \in P}(\delta - m^\top \Delta^\ell)_+ + \sum_{\ell \in I} |m^\top \Delta^\ell| + \mathbb{1}_{\mathcal{M}}(m) \\ g_2(z) \quad = \mathbb{1}_{\mathcal{Z}_\eta}(z) \end{cases}$$

with $\mathcal{Z}_\eta = \{z \in \mathbb{R}^d | z_i \geq \frac{\eta}{2}, i = 1, \ldots, d\}$. Functions g_1, g_2 are closed proper convex and sub-differentiable respectively over dom $g_1 = \mathcal{M}$ and dom $g_2 = \mathcal{Z}_\eta$, and f is convex and continuously differentiable over dom $g_1 \times$ dom g_2. Furthermore, for $j = 1, \ldots, d$, $(\nabla_1 f(m, z))_j = \lambda \frac{m_j}{z_j}$ and thus for any $m, z \in$ dom $g_1 \times$ dom g_2 and $d_1 \in \mathbb{R}^d$ such that $m + d_1 \in$ dom g_1 we have:

$$\|\nabla_1 f\,(m + d_1, z) - \nabla_1 f(m, z)\|_2 = \lambda \sqrt{\sum_{j=1}^{d} \frac{d_j^2}{z_j^2}} \leq \frac{2\lambda}{\eta} \|d_1\|_2 \tag{3.16}$$

Thus,$\nabla_1 f$ is uniformly Lispschitz continuous over dom g_1 with Lispschitz constant $L_1 = \frac{2\lambda}{\eta}$, while $\nabla_2 f$ is not uniformly Lispschitz continuous over dom g_2 , i.e., $L_2 = +\infty$. Therefore, Assumptions (i)-(iii) are satisfied and Problem $\mathcal{P}_\eta$ fits in the class of problems solvable by AM.

Let $m^{(k)}, z^{(k)}$ be the sequence generated by AM (see Algorithm 3.1). By Equation 3.9, we have $z_j^{(k)} = \sqrt{m_j^{(k)2} + \eta^2}, j = 1, \ldots, d$, and thus the AM algorithm yields the following IRLS sequence:

$$m^{(k+1)} \in \arg\min_{m} \sum_{\ell \in P}(\delta - m^\top \Delta^\ell)_+ + \sum_{\ell \in I} |m^\top \Delta^\ell| + \mathbb{1}_{\mathcal{M}}(m) + \frac{\lambda}{2} \sum_{j=1}^{d} \frac{m_j^2}{\sqrt{m_j^{(k)2} + \eta^2}}$$

Additionally, for any $m^{(0)} \in$ dom g_1, by Theorem 3.1 we have that for any $k \geq 2$:

$$H(m^{(k)}, z^{(k)}) - H^* \leq \max\left\{ \left(\frac{1}{2}\right)^{\frac{k-1}{2}} \left(H\left(m^{(0)}, z^{(0)}\right) - H^*\right), \frac{16\lambda R^2}{\eta(k-1)} \right\} \tag{3.17}$$

where H^* is the minimal value of H and R is a constant that only depends on H and the initialization of the sequence (see Theorem 3.1), and is thus independent of k. Finally, remark that $H^* = J_\eta^*$ and for any k, $H(m^{(k)}, z^{(k)}) = J_\eta(m^{(k)})$, and denote by C_k the *right-hand side of Equation 3.17. Then, using* $|x| \leq \sqrt{x^2 + \eta^2} \leq |x| + \eta$*, we obtain:*

$$\begin{aligned} J(m^{(k)}) - J^* &= J(m^{(k)}) - J_\eta(m^{(k)}) + J_\eta(m^{(k)}) - J_\eta^* + J_\eta^* - J^* \\ &\leq J_\eta(m^{(k)}) - J_\eta^* + \lambda d \eta \\ &\leq C_k + \lambda d \eta \end{aligned}$$

In the following, $\mathcal{P}_k$ refers to the problem solved at each iteration of the IRLS sequence (see Equation 3.15). Proposition 3.1 ensures that solving problems $\mathcal{P}_k$ for a sufficient number of iterations and a sufficiently small η provides a near-optimal solution to $\mathcal{P}$. The special interest of the IRLS method in our case is revealed when considering the dual formulation of each problem $\mathcal{P}_k$. Indeed, one can use *Lagrangian duality* to obtain a more compact mathematical programming formulation, as explained in the next Subsection.

2.2 Dual Formulation

2.2.1 Lagragian Duality

Lagrangian duality is a central concept in constrained optimization, where we consider optimization problems of the following form:

$$\begin{aligned} \min_{m \in \mathbb{R}^d} \; & f_0(m) \\ & f_i(m) \leq 0, \quad i = 1, \dots, s \\ & g_i(m) = 0, \quad i = 1, \dots, t \end{aligned} \tag{3.18}$$

where functions f_i, g_i are real-valued functions defined over $\mathbb{R}^d$. The *Lagragian* of Problem 3.18 is the function $\mathcal{L} : \mathbb{R}^d \times \mathbb{R}_+^s \times \mathbb{R}^t \to \mathbb{R}$, defined by:

$$\mathcal{L}(m, \alpha, \mu) = f_0(m) + \sum_{i=1}^{s} \alpha_i f_i(m) + \sum_{i=1}^{t} \mu_i g_i(m) \tag{3.19}$$

where $\alpha_i \geq 0$, (resp. μ_i) is referred to as the *Lagrange multiplier* associated with the i^{th} inequality (resp. equality) constraint. Intuitively, *the Lagrangian is used to formulate Problem 3.18 in an unconstrained form using penalties to penalize unfeasible solutions.* Indeed, remarking that $\max_{(\alpha,\mu) \in \mathbb{R}_+^s \times \mathbb{R}^t} \mathcal{L}(m, \alpha, \mu) = +\infty$ if there exists i such that $g_i(m) \neq 0$ or $f_i(m) > 0$ and $f_0(m)$ otherwise, Problem 3.18 reduces to:

$$\min_{m \in \mathbb{R}^d} \max_{(\alpha,\mu) \in \mathbb{R}_+^s \times \mathbb{R}^t} \mathcal{L}(m, \alpha, \mu)$$

The *Lagrange dual function*, denoted by g, is obtained by minimizing the Lagrangian w.r.t. the primal variable m, i.e., for any $(\alpha, \mu) \in \mathbb{R}_+^s \times \mathbb{R}^t$:

$$g(\alpha, \mu) = \min_{m \in \mathbb{R}^d} \mathcal{L}(m, \alpha, \mu) \tag{3.20}$$

Function g provides lower bounds on the optimal value of Problem 3.18. Indeed, for any $(\alpha, \mu) \in \mathbb{R}_+^s \times \mathbb{R}^t$, we have:

$$g(\alpha, \mu) \leq \min_{\substack{m \text{ s.t. } f_i(m) \geq 0, \forall i \\ g_i(m)=0, \forall i}} \mathcal{L}(m, \alpha, \mu) \leq \min_{\substack{m \text{ s.t. } f_i(m) \geq 0, \forall i \\ g_i(m)=0, \forall i}} f_0(m) \tag{3.21}$$

The *dual problem* (in contrast to Problem 3.18 referred to as the *primal problem*) can be regarded as the problem of finding the tightest lower bound and is thus formulated as the maximization of the dual function, i.e.,:

$$\max_{(\alpha,\mu) \in \mathbb{R}_+^s \times \mathbb{R}^t} g(\alpha, \mu) = \max_{(\alpha,\mu) \in \mathbb{R}_+^s \times \mathbb{R}^t} \min_{m \in \mathbb{R}^d} \mathcal{L}(m, \alpha, \mu) \tag{3.22}$$

Let us denote by g^* and f_0^* the optimal values of the dual Problem 3.22 and the primal Problem 3.18 respectively. Then by Equation 3.21, the *weak duality* property always holds, i.e., $g^* \leq f_0^*$. Furthermore, when Problem 3.18 is a *convex* optimization problem, (i.e., functions $f_i, i = 0, \ldots, s$ are convex and $g_i, i = 1, \ldots, t$ are linear functions) and the *Slater's condition* hold, then the *strong duality* property holds, i.e., $g^* = f_0^*$ (for the proof see §5.3.2 in [Boyd and Vandenberghe, 2004]).

Remark 3.2 (Slater's conditions). If $\mathcal{D}$ denotes the intersection of the domains of functions $f_i, i = 0, \ldots, s$, the *Slater's condition* requires that there exists a point m in the *relative interior*[1] of $\mathcal{D}$ such that the inequality constraints are strictly satisfied, i.e., $f_i(m) < 0, i = 1, \ldots, s$. When the inequality constraints are linear, the requirement reduces to $f_i(m) \leq 0, i = 1, \ldots, s$ which is equivalent to feasibility. Hence, when the objective function is convex and all inequality constraints are linear (or there is only equality constraints), strong duality always holds.

Remark 3.3 (convexity of the dual). It is also important to note that the dual problem is always convex, regardless of the convexity of the primal problem. Indeed, it can easily be checked that any pointwise minimum of an affine function, i.e., function of the form $h(\alpha) = \min_x \alpha^\top x$ is concave.

[1] The relative interior of convex set S is the set $\{m \in S | \exists \epsilon > 0 \text{ such that } B_\epsilon(m) \cap \text{aff}(S) \subseteq S\}$ where $\text{aff}(S)$ is the affine hull of S, and $B_\epsilon(m)$ is a ball of radius ϵ centered on m.

Therefore, under strong duality, it is equivalent to solve the dual or the primal problem. As the dual problem only involves sign constraints, it is sometimes easier to solve than the constrained primal Problem 3.18. Below, we recall necessary and sufficient optimality conditions for convex optimization problems, referred to as *Karush-Kuhn-Tucker (KKT) conditions*:

Theorem 3.2 (for the proof see §5.5.3 in [Boyd and Vandenberghe, 2004]). *Assume that $f_i, i = 0, \dots, s$ are convex differentiable functions and $g_i, i = 1, \dots, t$ are linear functions. If $m^*, (\alpha^*, \mu^*)$ satisfy the following conditions:*

- *(Primal and dual feasibility)* $\begin{cases} f_i(m^*) \leq 0, \ \alpha_i^* \geq 0, i = 1, \dots, s \\ g_i(m^*) = 0, \qquad\qquad i = 1, \dots, t \end{cases}$
- *(Complementary slackness)* $\alpha_i^* f_i(m^*) = 0, \ i = 1, \dots, s$
- *(Stationarity)* $\nabla_m \mathcal{L}(m^*, \alpha^*, \mu^*) = \mathbf{0}$

$$\Leftrightarrow \nabla f_0(m^*) + \textstyle\sum_{i=1}^{s} \alpha_i^* \nabla f_i(m^*) + \sum_{i=1}^{t} \mu_i^* \nabla g_i(m^*) = \mathbf{0}$$

then, m^* and (α^*, μ^*) are respectively the optimal solutions of the primal Problem 3.18 and dual Problem 3.22. Conversely, if $m^*, (\alpha^*, \mu^*)$ are optimal solutions, they necessary verify the conditions given above.

For more in-depth results and proofs, the reader may refer to [Bertsekas, 1997] or [Boyd and Vandenberghe, 2004] (Chapter 5). In the following, we illustrate these results on *support vector machines*, where the learning problem closely resembles the sub-problems $\mathcal{P}_k$ the IRLS sequence.

2.2.2 Support Vector Machine

In Subsection 3.1.2 of Chapter 1, we introduced the linear *support vector machine* (SVM) algorithm, which consists of learning a binary classifier of the form sign$(h(x))$ with a linear decision boundary, i.e., $h(x) = m^\top x + b$, $(m, b) \in \mathbb{R}^n \times \mathbb{R}$. However, the data may not be linearly separable, requiring the use of a non-linear classifier. In such cases, the input data can be projected from the original feature space $\mathcal{X}$ into a richer feature space $\mathcal{H}$ using a non-linear mapping function $\phi : \mathcal{X} \rightarrow \mathcal{H}$ [Cortes and Vapnik, 1995, Schiilkop et al., 1995, Cristianini and Shawe-Taylor, 2000]. If d denotes the dimension of $\mathcal{H}$, this yields the non-linear decision boundary $h(x) = m^\top \phi(x) + b$, $(m, b) \in \mathbb{R}^d \times \mathbb{R}$. Using the latter separating function, and a dataset $\{(x^\ell, y^\ell)\}_{\ell=1}^{t}$, where $x^\ell \in \mathcal{X} = \mathbb{R}^n$ and

$y^\ell \in \{-1, 1\}$, the SVM learning problem (see Problem 1.10) reformulates as follows:

$$\begin{aligned}
\min_{m\in\mathbb{R}^d, b\in\mathbb{R}, \epsilon\in\mathbb{R}^t_+} \quad & \sum_{\ell=1}^{t} \epsilon_\ell + \frac{\lambda}{2}\|m\|_2^2 \\
& y^\ell(m^\top \phi(x^\ell) + b) \geq 1 - \epsilon_\ell, \ell = 1, \dots, t \\
& \epsilon_\ell \geq 0, \quad \ell = 1, \dots, t
\end{aligned} \tag{3.23}$$

Let ϵ be the vector of slack variables $(\epsilon_1, \dots, \epsilon_t)$, then the *Lagrangian function* of Problem 3.23 can be derived by introducing Lagrange multipliers $(\alpha, \beta) \in \mathbb{R}^t_+ \times \mathbb{R}^t_+$:

$$\mathcal{L}(m, b, \epsilon, \alpha, \beta) = \sum_{\ell=1}^{t} \epsilon_\ell + \frac{\lambda}{2}\|m\|_2^2 - \sum_{\ell=1}^{t} \alpha_\ell \left(y^\ell(m^\top \phi(x^\ell) + b) - (1 - \epsilon_\ell)\right) - \sum_{\ell=1}^{t} \beta_\ell \epsilon_\ell$$

Then, the stationarity KKT condition (see Theorem 3.2) gives:

$$\nabla_m \mathcal{L}(m, b, \epsilon, \alpha, \beta) = \lambda m - \sum_{\ell=1}^{t} \alpha_\ell y^\ell \phi(x^\ell) = \mathbf{0} \Longrightarrow m = \frac{1}{\lambda}\sum_{\ell=1}^{t} \alpha_\ell y^\ell \phi(x^\ell) \tag{3.24}$$

$$\nabla_b \mathcal{L}(m, b, \epsilon, \alpha, \beta) = -\sum_{\ell=1}^{t} \alpha_\ell y^\ell = 0 \tag{3.25}$$

$$\nabla_\epsilon \mathcal{L}(m, b, \epsilon, \alpha, \beta) = \mathbf{1} - \alpha - \beta = 0 \tag{3.26}$$

Finally, substituting these conditions back into the Lagrangian gives the following dual problem :

$$\begin{aligned}
\max_{\alpha\in\mathbb{R}^t} & \sum_{\ell=1}^{t} \alpha_\ell - \frac{1}{2\lambda} \sum_{\ell,j=1}^{t} \alpha_\ell \alpha_j y^\ell y^j \phi(x^\ell)^\top \phi(x^j) \\
& \sum_{\ell=1}^{t} \alpha_\ell y^\ell = 0, \quad \mathbf{0} \leq \alpha \leq \mathbf{1}
\end{aligned} \tag{3.27}$$

This optimization problem is a convex quadratic program with t variables and a unique constraint (expected box constraints). Therefore, it is often computationally lighter than the primal Problem 3.23 when the number of examples t is low in front of d. Furthermore, the learning task can be entirely independent of d in the case where the inner products $\phi(x^\ell)^\top \phi(x^j)$ can be evaluated without explicitly computing the d-dimensional vectors $\phi(x)$. This trick is referred to as the *kernel trick* as the function $\kappa : (x, x') \mapsto \phi(x)^\top \phi(x')$ is called a *kernel function* [Cristianini and Shawe-Taylor, 2000, Schölkopf, 2002, Shawe-Taylor et al., 2004]. A basic example is the quadratic polynomial kernel $\kappa(x, x') = (x^\top x' + c)^2$ whose computation requires a number of operation in $O(n)$, whereas the attached projection function $\phi(x) = (x_1^2, \dots, x_n^2, \sqrt{2}x_n x_{n-1}, \dots, \sqrt{2}x_n x_1, \dots, \sqrt{2}x_2 x_1,$

$\sqrt{2}x_n c, \ldots, \sqrt{2}x_1 c, c)$ is of size $d = \frac{(n+2)(n+1)}{2}$. A more detailed overview of kernel-based machine learning methods will be given in Chapter 4.

Kernel SVMs have been leveraged for learning utility functions from preference examples on pairs of alternatives [Chapelle and Harchaoui, 2004, Radlinski and Joachims, 2005, Waegeman et al., 2009, Domshlak and Joachims, 2012], and in particular for learning Choquet integrals [Tehrani et al., 2014b, Tehrani, 2021] using the projection function $\phi(x) = (\phi_S(x_S))_{S \subseteq N}$ and $\phi_S(x_S) = \min_{i \in S}\{x_i\}$, since in this case we indeed have $C_w(x) = m^\top \phi(x)$ if m is the Möbius transform of w. Here, we show that similarly to a SVM, the subproblem $\mathcal{P}_k$ of the IRLS sequence given by Proposition 3.1, admits a compact dual formulation.

2.2.3 Dual Formulation of the IRLS Sequence

The IRLS sequence given by Proposition 3.1, consists in solving Problem $\mathcal{P}_k$ at each iteration, which after linearization of the pref-hinge loss reads as follows:

$$(\mathcal{P}_k) \qquad \min_{m \in \mathbb{R}^{2^n - 1}} \sum_{\ell \in P} \epsilon_\ell + \sum_{\ell \in I} (\epsilon_\ell^- + \epsilon_\ell^+) + \frac{\lambda}{2} \sum_{j=1}^{2^n - 1} \frac{m_j^2}{\sqrt{(m_j^{(k)})^2 + \eta^2}}$$

$$\begin{aligned} & m^\top \Delta^\ell + \epsilon_\ell \geq \delta, \;\; \ell \in P \\ & m^\top \Delta^\ell + \epsilon_\ell^+ - \epsilon_\ell^- = 0, \;\; \ell \in I \\ & m^\top \mathbf{1} = 1 \\ & \epsilon_\ell \geq 0, \;\; \ell \in P, \quad \epsilon_\ell^+, \epsilon_\ell^- \geq 0, \;\; \ell \in I \end{aligned}$$

where we recall that $\Delta^\ell = \phi(x^\ell) - \phi(x'^\ell)$ with $\phi(x) = (\phi_S(x_S))_{S \subseteq N}$ and $\phi_S(x_S) = \min_{i \in S}\{x_i\}$ or $\phi_S(x_S) = \prod_{i \in S} x_i$.

Since $\mathcal{P}_k$ is a convex problem with linear constraints, by Remark 3.2, 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, where the number of preference (resp. indifference) examples is denoted by p (resp. q), i.e., $p = |P|$ (resp. $q = |I|$):

Proposition 3.2. *Problem $\mathcal{P}_k$ admits a dual formulation $\mathcal{D}_k$ which has $p+q+1$ variables and $2(p+q)$ box constraints:*

$$(\mathcal{D}_k) \qquad \max_{\Gamma = (\alpha, \beta, \mu) \in \mathbb{R}^{p+q+1}} -\frac{1}{2\lambda} \Gamma^\top Q^\top (\gamma_k * Q) \Gamma + \Gamma^\top L$$

$$\begin{aligned} \mathbf{0} \leq \alpha \leq \mathbf{1} \\ -\mathbf{1} \leq \beta \leq \mathbf{1} \end{aligned}$$

where γ_k *is a vector containing the reciprocals of the current weights in the* ℓ_2*-regularization, i.e.,* $\gamma_k = \left(\sqrt{(m_j^{(k)})^2 + \eta^2}\right)_{j=1}^{d}$ *with* $d = 2^n - 1$*. Also,* $Q = (\Delta_P, \Delta_I, \mathbf{1})$ *is a data-dependent matrix of size* $d \times (p + q + 1)$ *where* $\Delta_P = (\Delta^\ell)_{\ell \in P}$ *and* $\Delta_I = (\Delta^\ell)_{\ell \in I}$ *are matrices of size* $d \times p$ *and* $d \times q$ *respectively. Finally,* $L = (\delta \mathbf{1}, \mathbf{0}, 1) \in \mathbb{R}^{p+q+1}$*.*

Proof. For the sake of conciseness, we write $\mathcal{P}_k$ *in a vectorial form:*

$$(\mathcal{P}_k) \qquad \min_{m \in \mathbb{R}^d, (\epsilon, \epsilon^+, \epsilon^-) \in \mathbb{R}_+^{p+2q}} \quad \mathbf{1}^\top \epsilon + \mathbf{1}^\top (\epsilon^+ + \epsilon^-) + \frac{\lambda}{2} m^\top (\gamma_k^{-1} * m)$$

$$\Delta_P^\top m + \epsilon \geq \delta \mathbf{1} \tag{3.28}$$

$$\Delta_I^\top m + \epsilon^+ - \epsilon^- = \mathbf{0} \tag{3.29}$$

$$m^\top \mathbf{1} = 1 \tag{3.30}$$

$$\epsilon \geq \mathbf{0}, \ \epsilon^+ \geq \mathbf{0}, \ \epsilon^- \geq \mathbf{0} \tag{3.31}$$

where $\epsilon = (\epsilon_1, \ldots, \epsilon_p), \epsilon^+ = (\epsilon_1^+, \ldots, \epsilon_q^+), \epsilon^- = (\epsilon_1^-, \ldots, \epsilon_q^-)$ *and* $\gamma_k^{-1} = \left(1/\sqrt{m_j^{(k)2} + \eta^2}\right)_{j=1}^{d}$ *with* $d = 2^n - 1$*. To compute the dual problem we write the Lagrangian function using the Lagrange multipliers* $\alpha \in \mathbb{R}_+^p$*,* $\beta \in \mathbb{R}^q$*,* $\mu \in \mathbb{R}$*, and* $t \in \mathbb{R}_+^p, u \in \mathbb{R}_+^q, v \in \mathbb{R}_+^q$ *respectively associated to constraints (3.28),(3.29),(3.30) and (3.31). To simplify notations,* Γ *denotes the concatenation of variables* (α, β, μ)*. Then, we have:*

$$\begin{aligned} \mathcal{L}(m, \epsilon, \epsilon^+, \epsilon^-, \Gamma, t, u, v) = \mathbf{1}^\top \epsilon + \mathbf{1}^\top (\epsilon^+ + \epsilon^-) + \frac{\lambda}{2} m^\top (\gamma_k^{-1} * m) + \alpha^\top (\delta \mathbf{1} - \Delta_P^\top m - \epsilon) \\ - \beta^\top ({\Delta_I}^\top m + \epsilon^+ - \epsilon^-) - \mu (m^\top \mathbf{1} - 1) - t^\top \epsilon - u^\top \epsilon^+ - v^\top \epsilon^- \end{aligned} \tag{3.32}$$

From Theorem 3.2, optimal solutions of the dual and primal problems necessarily satisfy the stationarity KKT conditions, i.e.,:

$$\nabla_m \mathcal{L} = \lambda (\gamma_k^{-1} * m) - \Delta_P \alpha - \Delta_I \beta - \mu \mathbf{1} = \mathbf{0} \Longrightarrow m = \frac{1}{\lambda} \gamma_k * Q \Gamma \tag{3.33}$$

$$\nabla_{\epsilon^+} \mathcal{L} = \mathbf{1} - \beta - u = \mathbf{0} \tag{3.34}$$

$$\nabla_{\epsilon^-} \mathcal{L} = \mathbf{1} + \beta - v = \mathbf{0} \tag{3.35}$$

$$\nabla_{\epsilon} \mathcal{L} = \mathbf{1} - \alpha - t = \mathbf{0} \tag{3.36}$$

Introducing these equations in Equation (3.32), we obtain:

$$\mathcal{L}(m, \epsilon, \epsilon^+, \epsilon^-, \Gamma, t, u, v) = -\frac{1}{2\lambda} \Gamma^\top Q^\top (\gamma_k * Q) \Gamma + \Gamma^\top L$$

Therefore, the dual problem, denoted by $\mathcal{D}_k$, reads as follows:

$$(\mathcal{D}_k) \quad \max_{\Gamma=(\alpha,\beta,\mu)\in\mathbb{R}^{p+q+1},(t,u,v)\in\mathbb{R}_+^{p+2q}} -\frac{1}{2\lambda}\Gamma^\top Q^\top(\gamma_k * Q)\Gamma + \Gamma^\top L$$

$$\begin{aligned} &\mathbf{1} - \alpha - t = \mathbf{0} \\ &\mathbf{1} - \beta - u = \mathbf{0} \\ &\mathbf{1} + \beta - v = \mathbf{0} \\ &\alpha \geq \mathbf{0} \end{aligned}$$

Since t, u, v do not appear in the objective function, $\mathcal{D}_k$ can be expressed as an optimization involving only Γ as variables:

$$(\mathcal{D}_k) \qquad \max_{\Gamma=(\alpha,\beta,\mu)\in\mathbb{R}^{p+q+1}} -\frac{1}{2\lambda}\Gamma^\top Q^\top(\gamma_k * Q)\Gamma + \Gamma^\top L$$

$$\begin{aligned} \mathbf{0} \leq \alpha \leq \mathbf{1} \\ -\mathbf{1} \leq \beta \leq \mathbf{1} \end{aligned}$$

Then $\mathcal{D}_k$ is a concave quadratic optimization problem with $p+q+1$ variables and $2(p+q)$ box constraints.

Remark 3.4 (support vectors). From the complementary slackness KKT condition (see Theorem 3.2), optimal solutions of the dual and primal problems verify:

$$\begin{cases} \alpha_\ell(\delta - \epsilon_\ell - m^\top \Delta^\ell) = 0, \quad \ell = 1, \dots, p \\ t_\ell \epsilon_\ell = 0 \Leftrightarrow (1-\alpha_\ell)\epsilon_\ell = 0, \quad \ell = 1, \dots, p \\ 0 \leq \alpha_\ell \leq 1, \epsilon_\ell \geq 0, \quad \ell = 1, \dots, p \end{cases}$$

Then, for any preference example $\ell \in P$ (i.e., such that $x^\ell \succsim x'^\ell$), if the learned model predicts a strict preference i.e., $m^\top \Delta^\ell - \delta > 0 \Leftrightarrow m^\top \phi(x^\ell) > m^\top \phi(x'^\ell) + \delta$, then $\alpha_\ell = 0$. Therefore, the optimal dual variable α is a sparse vector whose non-null components correspond to the preference examples the learned model m is incompatible with (or for which it predicts indifference, up to a margin of δ). In the SVM context, these non-null dual variables are referred to as the *support vectors* (for instance see [Shawe-Taylor et al., 2004] Chapter 7).

Towards higher dimensions. For a high number of viewpoints n, the computation of the matrix $Q^\top(\gamma_k * Q)$ raises an issue since Q and γ_k have $d = 2^n - 1$ rows. More precisely, this dot product requires a number of operations in $O(2^n(p+q+1)^2)$, in addition to the computation of Q itself. However, at the first iteration of the IRLS sequence, the matrix $Q^\top(\gamma_k * Q)$ reduces to the *kernel matrix* associated to the mapping function $\tilde{\phi} : \mathcal{X}^2 \to \mathbb{R}$ such that for any pair $x, x' \in \mathcal{X}^2$, $\tilde{\phi}(x, x') = \phi(x) - \phi(x')$. Indeed, taking $m^{(0)} = \frac{1}{d}\mathbf{1}$ we have:

$$\begin{aligned} Q^\top(\gamma_0 * Q) &= \sqrt{d^{-2} + \eta^2} \begin{pmatrix} \Delta_P^\top \Delta_P & \Delta_P^\top \Delta_I & \Delta_P^\top \mathbf{1} \\ \Delta_I^\top \Delta_P & \Delta_I^\top \Delta_I & \Delta_I^\top \mathbf{1} \\ \mathbf{1}^\top \Delta_P & \mathbf{1}^\top \Delta_I & \mathbf{1}^\top \mathbf{1} \end{pmatrix} \\ &= \sqrt{d^{-2} + \eta^2} \begin{pmatrix} K_{PP} & K_{PI} & K_{P\mathcal{S}} \\ K_{IP} & K_{II} & K_{I\mathcal{S}} \\ K_{\mathcal{S}P} & K_{\mathcal{S}I} & K_{I\mathcal{S}} \end{pmatrix} \end{aligned} \tag{3.37}$$

where $\mathcal{S}$ is a set containing the index of the pair $(x^\ell, x'^\ell) = (\mathbf{1}, \mathbf{0})$, and for any index set $A_1, A_2 \in \{P, I, \mathcal{S}\}$, $K_{A_1 A_2}$ is a kernel matrix of size $|A_1| \times |A_2|$ such that:

$$(K_{A_1 A_2})_{\ell, j} = \kappa((x^\ell, x'^\ell), (x^j, x'^j)), \quad (\ell, j) \in A_1 \times A_2$$

with $\kappa : \mathcal{X}^2 \times \mathcal{X}^2 \to \mathbb{R}$ the kernel function attached to $\tilde{\phi}$, i.e.:

$$\begin{aligned} \kappa((x^\ell, x'^\ell), (x^j, x'^j)) &= \tilde{\phi}(x^\ell, x'^\ell)^\top \tilde{\phi}(x^j, x'^j) \\ &= \phi(x^\ell)^\top \phi(x^j) + \phi(x'^\ell)^\top \phi(x'^j) - \phi(x^\ell)^\top \phi(x'^j) - \phi(x'^\ell)^\top \phi(x^j) \end{aligned}$$

Remark 3.5 (preference kernel). The function κ assigns a measure of similarity to each pair of preference or indifference examples, (x^ℓ, x'^ℓ) and (x^j, x'^j), in the space induced by the projection function $\tilde{\phi}$. In other words, $\kappa((x^\ell, x'^\ell), (x^j, x'^j))$ increases as the difference between $\phi(x^\ell)$ and $\phi(x'^\ell)$ becomes closer to the difference between $\phi(x^j)$ and $\phi(x'^j)$, thus suggesting that the preference relation between x^ℓ and x'^ℓ should resemble that between x^j and x'^j. Such *preference kernel* can also be encountered in the SVM-based approaches for learning utility functions from pairwise comparisons [Waegeman et al., 2009, Domshlak and Joachims, 2012].

Hence, by Equation 3.37, the computation of the matrix $Q^\top(\gamma_0 * Q)$ solely involves the computation of inner products of the form $\phi(x)^\top \phi(x)$. Therefore, similarly as in the SVM *kernel trick* (see Subsection 2.2.2), we can exploit direct computations of the inner products. A computation in $O(n^2)$ is known for the case of the Choquet integral [Tehrani et al., 2014b], i.e., for $\phi_S(x_S) = \min_{i \in S}\{x_i\}, S \subseteq N$:

***Proposition 3.3** (see Section 3 in [Tehrani et al., 2014b]).*

$$\phi(x)^\top \phi(x) = x^\top x' + \sum_{i=1}^{n-1} x_{(i)} \left\{ \sum_{j=1}^{n-i} 2^{n-i-j} \cdot \min \left\{ x'_{(i)}, x'_{[j+1]_i} \right\} \right\}$$

where $(.)$ is a permutation of N such that $x_{(i)} \leq x_{(i+1)}$ and $[.]_i$ are permutations sorting each vector $(x'_{(i+1)}, \ldots, x'_{(n)})$ by increasing order.

This formula can also be used to obtain a polynomial computation of $\phi(x)^\top \phi(x)$ when $\phi_S(x_S) = \max_{i \in S}\{x_i\}$ since $\max_{i \in S}\{x_i\} = -\min_{i \in S}\{-x_i\}$. In addition, we provide a polynomial computation for the multilinear utility, i.e., for $\phi_S(x_S) = \prod_{i \in S} x_i,\ S \subseteq N$ in the following proposition:

***Proposition 3.4.** When $\phi_S(x_S) = \prod_{i \in S} x_i$, $S \subseteq N$, we have:*

$$\phi(x)^\top \phi(x) = \sum_{S \subseteq N} \prod_{i \in S} x_i \prod_{i \in S} x'_i = \prod_{i=1}^{n} (x_i x'_i + 1) - 1$$

Then $\phi(x)^\top \phi(x)$ can be computed in $O(n)$.

Proof. We provide a proof by induction. For $n = 1$, $\phi(x)^\top \phi(x) = x_1 x'_1 = \prod_{i=1}^{1} (x_i x'_i + 1) - 1$. Now let us assume that the property is valid for some $n \in \mathbb{N}$. For $n+1$, we have:

$$\begin{aligned}
\prod_{i=1}^{n+1} (x_i x'_i + 1) &= (x_{n+1} x'_{n+1} + 1) \prod_{i=1}^{n} (x_i x'_i + 1) \\
&= (x_{n+1} x'_{n+1} + 1) (\sum_{S \subseteq N} \prod_{i \in S} x_i \prod_{i \in S} x'_i + 1) \\
&= \sum_{S \subseteq N} \prod_{i \in S} x_{n+1} x_i \prod_{i \in S} x'_{n+1} x'_i + x_{n+1} x'_{n+1} + \sum_{S \subseteq N} \prod_{i \in S} x_i \prod_{i \in S} x'_i + 1 \\
&= \sum_{S \subseteq N} \prod_{i \in S \cup \{n+1\}} x_i \prod_{i \in S \cup \{n+1\}} x'_i + x_{n+1} x'_{n+1} + \sum_{S \subseteq N} \prod_{i \in S} x_i \prod_{i \in S} x'_i + 1 \\
&= \sum_{S \subseteq N \cup \{n+1\}} \prod_{i \in S} x_i \prod_{i \in S} x'_i + 1.
\end{aligned}$$

This kernel also appears in [Shawe-Taylor et al., 2004] (Chapter 9), under the name of all-subset kernel.

Taking into consideration these polynomial computations, we propose to proceed to a kernelized computation of matrix $Q^\top(\gamma_k * Q)$ for the first iteration of the IRLS sequence, yielding a number of operations in $O(n(p+q+1)^2)$ for the multilinear utility and in $O(n^2(p+q+1)^2)$ for the Choquet integral, instead of $O(2^n(p+q+1)^2)$ for the unkernelized version. This provides a way to perform a dimension reduction since non-significant coefficients (whose absolute values are lower than some threshold ν) obtained after this first iteration can be discarded before going on. It is important to note that this dimension reduction step requires evaluating the learned model in the primal space using Equation 3.33, i.e., computing $m^{(1)} = \frac{1}{\lambda}\gamma_0 * Q\Gamma$, yielding a *partial kernelization*, as Q, and thus the vectors $\phi(x)$, have to be explicitly computed. In the next section, we show that this partial kernelization is adequate for handling problems with more than 20 viewpoints, which, as far as we know, has never been done in the literature on capacity-based preference model learning.

Before that, we give the overall learning algorithm in Algorithm 3.2 where for the sake of clarity, the following notation is used for any matrix K:

$$\operatorname{sol}(K) := \underset{\Gamma \in [0,1]^p \times [-1,1]^q \times \mathbb{R}}{\arg\max} \; -\frac{1}{2\lambda}\Gamma^\top K \Gamma + \Gamma^\top L$$

Algorithm 3.2: Dual IRLS Algorithm

```
Inputs: $\mathcal{D} = \{(x^\ell, x'^\ell)\}_{\ell \in P \cup I \cup \mathcal{S}}, \kappa, \phi, \lambda, \eta, \epsilon, \nu, \delta, d$
// Initialization
$Q \leftarrow (\phi(x^\ell) - \phi(x'^\ell))_{\ell \in P \cup I \cup \mathcal{S}}, \;\; L \leftarrow (\delta\mathbf{1}, \mathbf{0}, 1)$
$k, \; m^{(0)}, \; \gamma_0 \leftarrow 0, d^{-1}\mathbf{1}, \sqrt{d^{-2} + \eta^2}\mathbf{1}$
// First iteration of the IRLS sequence
$K \leftarrow (\sqrt{d^{-2} + \eta^2}\kappa((x^\ell, x'^\ell), (x^j, x'^j)))_{\ell, j \in P \cup I \cup \mathcal{S}}$
$\Gamma^* \leftarrow \operatorname{sol}(K)$
$k, \; m^{(1)} \leftarrow 1, \; \frac{1}{\lambda}\gamma_0 * Q\Gamma^*$
// Dimension reduction
$\mathcal{A} \leftarrow \left\{ j \middle| \, |m_j^{(1)}| > \nu, \; j = 1, \dots, d \right\}$
$m^{(1)}, Q \leftarrow (m_1^{(j)})_{j \in \mathcal{A}}, (Q_j)_{j \in \mathcal{A}}$
$m^{(1)} \leftarrow m^{(1)} / \mathbf{1}^\top m^{(1)}$
// IRLS sequence
while $\|m^{(k)} - m^{(k-1)}\|_2 > \epsilon$ do
    $\gamma_k \leftarrow (\sqrt{(m_j^{(k)})^2 + \eta^2})_{j \in \mathcal{A}}$
    $K \leftarrow Q^\top(\gamma_k * Q)$
    $\Gamma^* \leftarrow \operatorname{sol}(K)$
    $m^{(k+1)} \leftarrow \frac{1}{\lambda}\gamma_k * Q\Gamma^*$
    $k \leftarrow k + 1$
Outputs: $m^{(k)}$
```

Remark 3.6 (dimension reduction). Performing an initial coefficient selection after the first iteration appears to be a reasonable option, as the first iteration corresponds to the ℓ_2-regularized version of the learning problem $\mathcal{P}$. Indeed, while being unable to provide sparse solution, the ℓ_2-regularization is known to successfully help capturing the underlying structure of the data. In the context of linear regression, the ℓ_2-regularized solution is viewed as a reliable witness of the relative coefficients importance and is used to weight the ℓ_1-regularization in the *adaptive LASSO* [Zou, 2006] to improve coefficient selection in the presence of correlated features. This behavior was also demonstrated in practice for the learning of Choquet integrals (in Chapter 2, see Section 2.2.3 for the formulation of the adaptive ℓ_1-regularized preference learning problem and Section 3 for numerical experiments). However, a too large threshold hyperparameter ν could obviously prematurely exclude important coefficients. On the other side, a too small threshold could yield too large matrices Q. Therefore, in practice, we select the value of ν leading to an optimal tradeoff between training time and test error (using cross-validation).

For the sake of completeness, we also provide below a variant of the algorithm in the regression setting.

Counterpart algorithm in the regression setting In what follows, we consider a dataset of alternatives and overall evaluations $\{x^\ell, y^\ell\}_{\ell=1}^t$ with $x^\ell \in \mathcal{X}, y^\ell \in \mathbb{R}$. In this case, the initial learning problem can be formulated as follows using the δ-insensitive loss for any $\delta \geq 0$ (see also Subsection 2.1.2 in Chapter 4):

$$(\mathcal{P}) \min_{m \in \mathbb{R}^{2^n-1}} \sum_{\ell=1}^{t} [\delta - |m^\top \phi(x^\ell) - y^\ell|]_+ + \lambda \|m\|_1 \tag{3.38}$$

Then, the regression errors can be linearized using auxiliary variables $\epsilon_\ell^+, \epsilon_\ell^- \geq 0, \ell = 1, \dots, t$ as follows:

$$\begin{aligned}(\mathcal{P}) \min_{m \in \mathbb{R}^{2^n-1}, \epsilon^+ \in \mathbb{R}_+^t, \epsilon^- \in \mathbb{R}_+^t} \quad & \sum_{\ell=1}^{t} (\epsilon_\ell^+ + \epsilon_\ell^-) + \lambda \|m\|_1 \\ & y^\ell - m^\top \phi(x^\ell) \leq \delta + \epsilon_\ell^+, \quad \ell = 1, \dots, t \\ & m^\top \phi(x^\ell) - y^\ell \leq \delta + \epsilon_\ell^-, \quad \ell = 1, \dots, t\end{aligned} \tag{3.39}$$

Then, Proposition 3.1 can be readily adapted to show that, similarly to the preference setting, by combining the quadratic variational formulation of the ℓ_1-norm (see Equation 3.9) with the alternating minimization algorithm (see Algorithm 3.1), we obtain a sequence of least-squares problems defined for any $k \geq 0$ and $\eta > 0$ as follows:

$$(\mathcal{P}_k) \qquad \min_{m\in\mathbb{R}^{2^n-1},\epsilon^+\in\mathbb{R}^t_+,\epsilon^-\in\mathbb{R}^t_+} \sum_{\ell=1}^{t}(\epsilon_\ell^- + \epsilon_\ell^+) + \frac{\lambda}{2}\sum_{j=1}^{2^n-1}\frac{m_j^2}{\sqrt{(m_j^{(k)})^2+\eta^2}}$$
$$y^\ell - m^\top\phi(x^\ell) \leq \delta + \epsilon_\ell^+, \quad \ell = 1,\dots,t$$
$$m^\top\phi(x^\ell) - y^\ell \leq \delta + \epsilon_\ell^-, \quad \ell = 1,\dots,t$$

$\mathcal{P}_k$ coincides with a *support vector regression* problem (i.e., the regression counterpart of SVM, described in more detail in Subsection 2.1.2 of Chapter 4), with a weighted ℓ_2-regularization using the weights $\gamma_k^{-1} = \left(1/\sqrt{m_j^{(k)2}+\eta^2}\right)_{j=1}^{d}$, where $d = 2^n - 1$. Then, it can easily be checked that $\mathcal{P}_k$ admits the following dual formulation:

$$\max_{\mu^+,\mu^-\in[0,1]^t} -\frac{1}{2\lambda}(\mu^+-\mu^-)^\top K(\mu^+-\mu^-) + Y^\top(\mu^+-\mu^-) - \delta\mathbf{1}^\top(\mu^++\mu^-) \tag{3.40}$$

with $K = \Phi(\gamma_k * \Phi^T) \in \mathbb{R}^{t\times t}$, $\Phi = (\phi(x^\ell))_{\ell=1}^t \in \mathbb{R}^{t\times d}$, and $Y = (y^\ell)_{\ell=1}^t \in \mathbb{R}^t$.

For $k = 0$, taking $m^{(0)} = \frac{1}{d}\mathbf{1}$, we have: $K = \Phi(\gamma_0 * \Phi^T) = \sqrt{d^{-2}+\eta^2}(\kappa(x^\ell, x'^\ell))_{\ell,\ell'=1}^t$ with $\kappa(x,x') = \phi(x)^T\phi(x')$ that can be computed in polynomial time in n with Equation 3.3 or Equation 3.4 depending on the chosen interaction function ϕ, for any $x, x' \in \mathcal{X}$. Finally, let sol(K) denotes the solution of Problem 3.40 for any kernel matrix K, then the D-IRLS algorithm for the regression setting is given in Algorithm 3.3

Enforcing monotonicity. In the initial learning problem $\mathcal{P}$ (see Section 1 for the preference setting or Problem 3.38 for the regression setting), monotonicity constraints have been omitted. However, 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 [Tehrani, 2021] where the learned models achieve low monotonicity violation rates when the training data does not violate monotonicity. 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.

Recall that the monotonicity of any capacity w can be guaranteed by asking that for any viewpoint coalition $S \subseteq N$, removing a viewpoint $i \in S$ cannot increase the capacity value, i.e., $w(S) \geq w(S \setminus \{i\})$. These constraints translate in terms of the Möbius transform m by $\sum_{T\subseteq S, T\ni i} m(T) \geq 0, \;\; \forall i \in S, \forall S \subseteq N$, using $w(S) = \sum_{T\subseteq S} m(T)$. Therefore, if $C(n)$ denotes the number of monotonicity constraints for n viewpoints, we

Algorithm 3.3: Dual IRLS algorithm in the regression setting

```
Inputs: $\mathcal{D} = \{(x^\ell, y^\ell)\}_{\ell=1}^t, \kappa, \phi, \lambda, \eta, \epsilon, \nu, \delta, d$
// Initialization
$\Phi \leftarrow (\phi(x^\ell))_{\ell=1}^t, Y = (y^\ell)_{\ell=1}^t$
$k,\ m^{(0)},\ \gamma_0 \leftarrow 0, d^{-1}\mathbf{1}, \sqrt{d^{-2} + \eta^2}\mathbf{1}$
// First iteration of the IRLS sequence
$K \leftarrow (\sqrt{d^{-2} + \eta^2}\kappa(x^\ell, x'^\ell))_{\ell,\ell'=1}^t$
$\mu^+, \mu^- \leftarrow \mathrm{sol}(K)$
$k,\ m^{(1)} \leftarrow 1,\ \frac{1}{\lambda}\gamma_0 * \Phi^T(\mu^+ - \mu^-)$
// Dimension reduction
$\mathcal{A} \leftarrow \left\{ j \middle| \, |m_j^{(1)}| > \nu,\ j = 1, \ldots, d \right\}$
$m^{(1)}, \Phi \leftarrow (m_1^{(j)})_{j\in\mathcal{A}}, (\Phi^j)_{j\in\mathcal{A}}$
$m^{(1)} \leftarrow m^{(1)}/\mathbf{1}^\top m^{(1)}$
// IRLS sequence
while $\|m^{(k)} - m^{(k-1)}\|_2 > \epsilon$ do
    $\gamma_k \leftarrow (\sqrt{(m_j^{(k)})^2 + \eta^2})_{j\in\mathcal{A}}$
    $K \leftarrow \Phi(\gamma_k * \Phi^T)$
    $\mu_+, \mu_- \leftarrow \mathrm{sol}(K)$
    $m^{(k+1)} \leftarrow \frac{1}{\lambda}\gamma_k * \Phi^T(\mu^+ - \mu^-)$
    $k \leftarrow k + 1$
Outputs: $m^{(k)}$
```

have:

$$C(n) = \sum_{k=1}^{n} k \binom{n}{k} \tag{3.41}$$

Including in $\mathcal{P}$ this set of constraints induces a dual problem $\mathcal{D}_k$ with $p+q+1+C(n)$ variables, since each constraint of the primal induces a dual variable. Thus the dualization benefit is lost and one may prefer a direct solving of $\mathcal{P}$ with linear programming (LP), as proposed in Subsection 2.2.3 of Chapter 2. Still, the exponential number of variables and constraints is an obstacle to scalability. Hence we propose to handle the monotonicity constraints throughout a *constraint generation* algorithm (also known as *cutting-plane* 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 $\mathcal{P}$ found without monotonicity constraints. Then, at each iteration, if the current solution does not verify monotonicity constraints, a violated constraint is inserted in $\mathcal{P}$, and the problem is resolved again. If, however, the current solution does verify monotonicity constraints, then the current solution is the optimal solution of the fully constrained optimization problem and the algorithm stops. This iterative procedure is formally given in Algorithm 3.4, where C_k refers to the index set of monotonicity constraints inserted up to iteration k and $\mathcal{P}_{C_k}$ is the optimization problem $\mathcal{P}$ with the inserted constraints.

Algorithm 3.4: Constraint Generation Algorithm

```
$C_0 \leftarrow \emptyset$
$m^{(0)} \leftarrow$ solution of $\mathcal{P}_{C_0}$ (obtained with LP)
while $m^{(k)}$ does not verify monotonicity constraints do
    $c \leftarrow$ index of a violated constraint
    $C_{k+1} \leftarrow C_k \cup \{c\}$
    $m^{(k+1)} \leftarrow$ solution of $\mathcal{P}_{C_k}$ (obtained with LP)
    $k \leftarrow k + 1$
Outputs: $m^{(k)}$
```

The next section presents numerical evidence of the benefits of the Dual IRLS method (Algorithm 3.2) when monotonicity constraints are relaxed, and of the constraint generation algorithm (Algorithm 3.4) when monotonicity constraints are enforced

3 Numerical Tests

3.1 Synthetic Preference Data

In this subsection we present the results of numerical tests performed on synthetic preference data. We first test the ability of the dual IRLS method (see Algorithm 3.2), denoted by D-IRLS, to learn a multilinear utility or a Choquet integral for a growing number of viewpoints. We compare it to an exact solving of $\mathcal{P}$ with LP (see Problem 3.8 for the linearized problem), denoted by ES. Preference data is generated using the process detailed in Subsection 3.1 of Chapter 2 (see Paragraph *Data generation for learning the capacity*). We set the size of the training sets to $p = q = 250$ and of the test sets to $p = 1000$ and $q = 0$. The generalizing performances of the learned models are assessed with the *test error*, computed here as the proportion of inverted preferences in the test set.

The linear and quadratic optimization tasks are conducted using the mathematical programming Gurobi solver (version 9.1.2) on a 2.8 GHz Intel Core i7 processor with 16GB RAM. For both learning methods, the ℓ_1-norm regularization parameter λ is set to $\lambda = 1$. For the D-IRLS method, the smoothing parameter is set to $\eta = 10^{-50}$, the termination parameter is $\epsilon = 10^{-3}$ and the thresholding parameter is set to $\nu = 10^{-5}$.

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

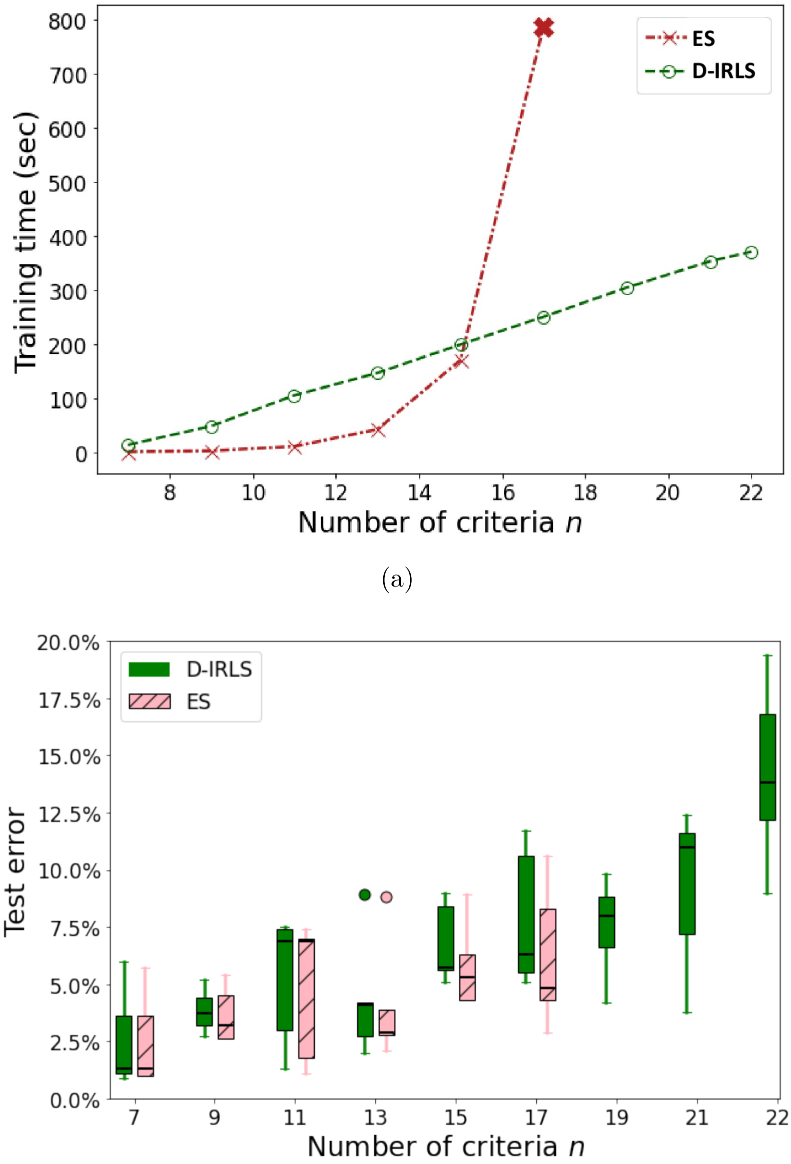
n	$\tilde{C}(n)$	$C(n)$	Time ESG	Time ESC
6	**3.2±6.4**	192	0.6±0.2	**0.6±0.1**
9	**2.4±7.2**	2304	**4.2±1.9**	18.0±4.6
12	**151.9±222.2**	24576	**61.0±30.4**	1212.6±247.6
15	**2777.6±4326.5**	245760	**3448.6±5613.1**	-

Table 3.1: $C(n)$,$\tilde{C}(n)$ and training times (sec.) for ESG and ESC.

the generalizing performances of the learned models. In order to evaluate the scalability of our method we vary the number of viewpoints from $n = 7$ to $n = 22$. Figure 3.2 (resp. Figure 3.3) shows the results for the learning of the Choquet integral in its conjunctive form (resp. of the multilinear utility). More precisely, in Figure 6.2,6.4, are represented for both models the average training times, in red for ES and green for D-IRLS, while we show the test error in Figure 6.3,6.5, also in green for D-IRLS and in pink for ES. 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 viewpoints for both methods. Finally, we can notice that the test errors obtained for the learning of the multilinear utility are higher than the ones obtained for the learning of the Choquet integral.

Enforcing monotonicity In a second experiment, we assess the computational efficiency of the constraint generation algorithm (see Algorithm 3.4) 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 $\mathcal{P}$ under all monotonicity constraints (denoted ESC) with the exact solving of $\mathcal{P}$ with constraint generation (denoted ESG). Both are solved using LP. In Table 3.1 we compare $C(n)$ the total number of monotonicity constraints in ESC, and $\tilde{C}(n)$ the average number of constraints generated in ESG. The best results are highlighted in bold. We observe that ESC (including all constraints) is slower for $n = 6, 9, 12$ than ES and limited to $n = 12$. ESG performs significantly better (up to 15 viewpoints) 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.

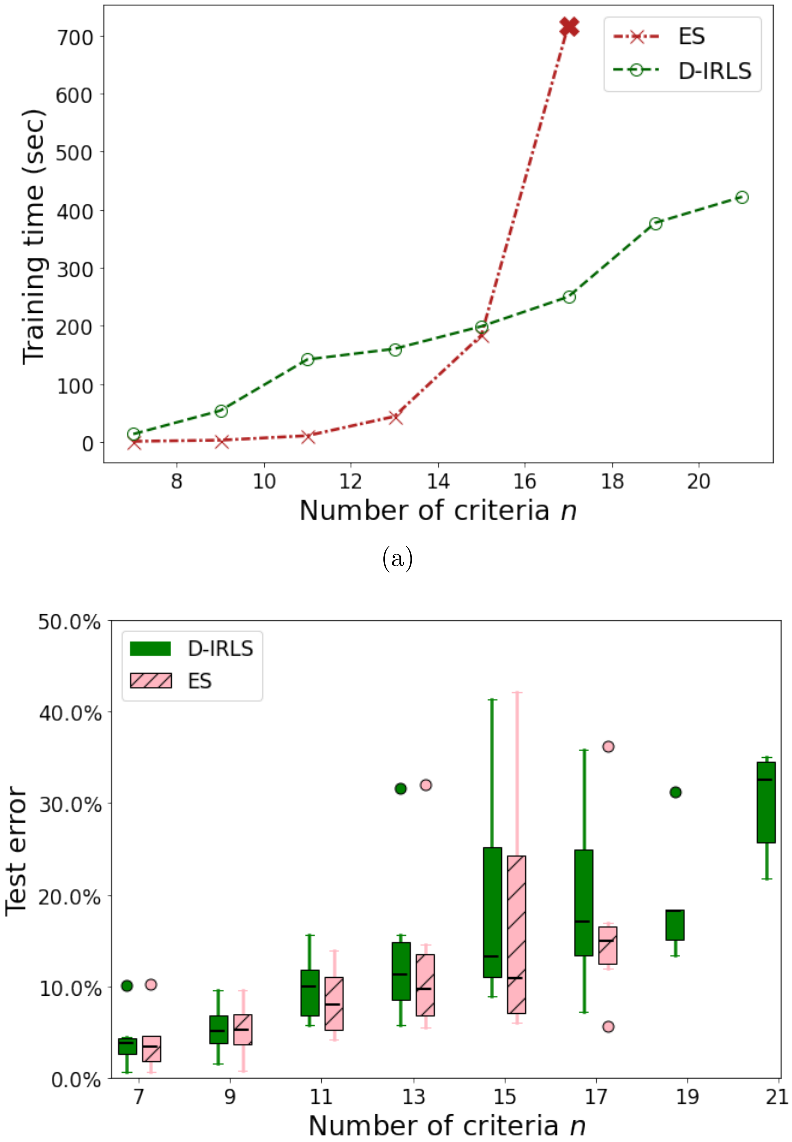
Comparison with k-additive models. The advantage of using sparse models with possible large interactions instead of k-additive models is illustrated in Table 3.2 where



(a)

(b)

Figure 3.2: Training time (avg.) and test error (boxplot) for D-IRLS and ES with the Choquet Integral.



(a)

(b)

Figure 3.3: Training time (avg.) and test error (boxplot) for D-IRLS and ES with the multilinear utility.

we compare our method (D-IRLS) to an exact solving of $\mathcal{P}$ with k-additivity constraints (denoted by k-add) for $k = 2$ and $k = 3$, still under the same experimental setting. With D-IRLS, the generalizing performance is significantly improved compared to k-add, while computation times remain admissible for $n \leq 12$, and gets better for larger n (i.e., $n \geq 16$).

Mixing different models of interactions. The proposed methods allow learning an instance of model F_m defined by Equation 6.1, with a chosen interaction function ϕ_S defining the nature of interaction terms, that may be the $\min, \max$, product, or any

	Test Error			Training time		
n	D-IRLS	2-add	3-add	D-IRLS	2-add	3-add
8	**0.04**	0.10	0.06	28.22	**1.33**	1.40
12	**0.04**	0.17	0.23	122.53	**20.96**	21.34
16	**0.06**	0.22	0.49	**187.79**	345.13	346.78

Table 3.2: Average test error and training time of D-IRLS in comparison to k-add models.

given monotonic function. However, several interaction functions may coexist in the same preference model, and thus allowing for different types of interactions in F_m could provide a benefit in terms of sparsity of the learned representations.

As an illustration, we consider the *Hurwicz criterion* [Hurwicz, 1951], which is standardly used to make a tradeoff between the worst and the best components, i.e.,

$$h(x) = \alpha \min_{i \in N}\{x_i\} + (1-\alpha) \max_{i \in N}\{x_i\}, \qquad 0 \leq \alpha \leq 1 \tag{3.42}$$

Although the term $\min_{i \in N}\{x_i\}$ (resp. $\max_{i \in N}\{x_i\}$) term in h admits a sparse representation in the F_m model using the interaction function $\phi_S(x_S) = \min_{i \in S}\{x_i\}$ (resp. $\phi_S(x_S) = \max_{i \in S}\{x_i\}$), this is not the case of h that includes both terms. This suggests extending the model F_m defined in Equation 6.1 to include simultaneously several instances of ϕ_S (like min and max).

The Choquet integral already provides a framework for such modeling. Indeed, if we write $w = w^{\wedge} + w^{\vee}$ with $(w^{\wedge}, w^{\vee})$ two sub-normalized capacities (i.e., such that $w^{\wedge}(N) + w^{\vee}(N) = 1$), we have $C_w(x) = C_{w^{\wedge}}(x) + C_{w^{\vee}}(x)$. Then, using the conjunctive form of the Choquet integral (see Equation 3.2) for $C_{w^{\wedge}}(x)$ and the disjunctive form (see Equation 3.3) for $C_{w^{\vee}}(x)$, C_w reads as a sum of interaction terms with two types of interactions:

$$C_w(x) = \sum_{S \subseteq N} (m_{w^{\wedge}}(S) \min_{i \in S}\{x_i\} + m_{\bar{w}^{\vee}}(S) \max_{i \in S}\{x_i\}) \tag{3.43}$$

This formulation deliberately includes redundancy terms to facilitate the emergence of sparse formulations. For instance, the Hurwicz model h of Equation 3.43 is a particular case of Equation 3.43 with only two non-null coefficients, $m_{w^{\wedge}}(N)$ and $m_{\bar{w}^{\vee}}(N)$.

Using this formulation, the proposed learning method can be adapted to obtain a sparse representation of C_w possibly including both conjunctive and disjunctive terms. To this end, we solve a variant of problem $\mathcal{P}$ using the double Möbius vector $(m^{\wedge}, m^{\vee})$ and the double ℓ_1-regularization term $\lambda_{\wedge}\|m^{\wedge}\|_1 + \lambda_{\vee}\|m^{\vee}\|_1$ under the normalization constraint

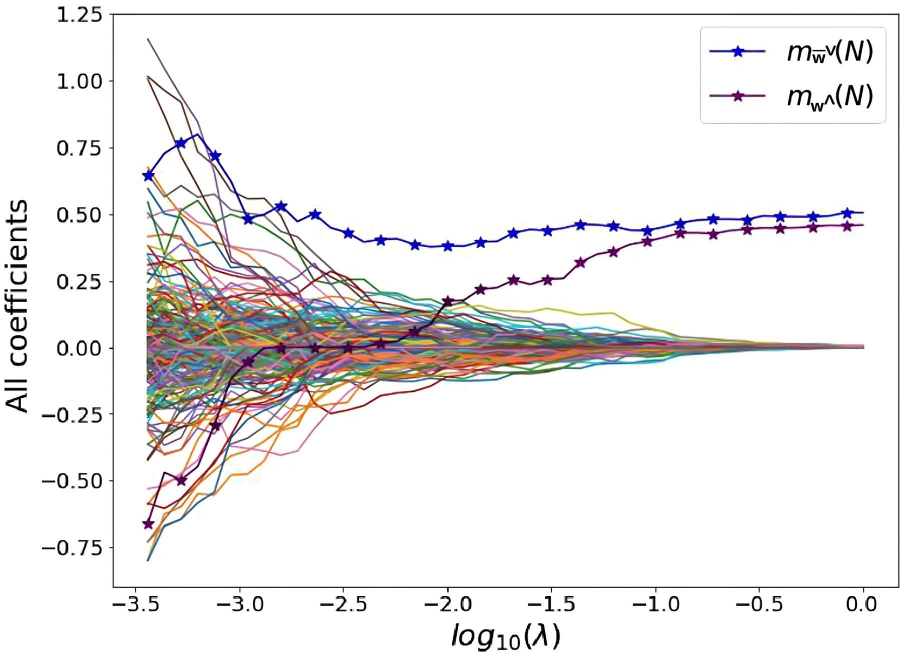


Figure 3.4: Selection path for the learning of the Hurwicz model.

$(m^\wedge + m^\vee)^\top \mathbf{1} = 1$.

This variant is evaluated on synthetic preference data generated using the Hurwicz model for $\alpha = 0.5$ (see Equation 3.42) for $n = 8$. In Figure 3.4 we provide the *regularization path* obtained for an increasing level of regularization, i.e., we represent the learned coefficient values $(m_j^\wedge, m_j^\vee), j = 1, \ldots, 2^n - 1$ w.r.t. to the regularization hyperparameter $\lambda = \lambda_\wedge = \lambda_\vee$. The non-null ground truth coefficients attached to the $\min_{i \in N}\{x_i\}$ and $\max_{i \in N}\{x_i\}$ terms (i.e., $m_{w^\wedge}(N)$ and $m_{\bar{w}^\vee}(N)$) are highlighted with star markers. As expected, a model including only these two factors is progressively emerging with the increase of the regularization level.

Besides, this model clearly illustrates the idea that considering only small interactions (with k-additivity constraints) is limiting since h presents interaction terms involving the entire set of viewpoints and cannot be simply approximated by interactions on smaller sets.

3.2 Application to Judicial Decision-Making in Divorce Cases

In this section, we focus on a particular application case that is the problem of predicting the compensatory allowance ("prestation compensatoire" in French) set by the judge in divorce proceedings [2]. The compensatory allowance (CA) is an amount

[2] This case study is a collaborative work with Fabien Tarissan, Isabelle Sayn and Patrice Perny, which was the subject of the presentation *Leveraging the Choquet Integral for Analyzing Court Decisions in Divorce Cases* at ESELS 2025 (European Society for Empirical Legal Studies https://esels.eu/).

intended, according to Article 270 of the French Civil Code, to "compensate, as far as possible, for the disparity that the breakdown of the marriage creates in the respective living conditions." This amount, expressed in euros, must be paid either as a lump sum or as a regular annuity by one spouse to the other. When the spouses fail to reach an agreement on the amount, it is then up to the judge to determine it based on their assessment of the disparity in living standards that the dissolution of the marriage is likely to cause between the spouses.

This decision-making task is known to be challenging, notably due to the ambiguity of the Civil Code regarding how the amount should be determined. The Code sets out a general principle: "The compensatory allowance is determined based on the needs of the spouse to whom it is paid and the resources of the other, taking into account the situation at the time of the divorce and how it may evolve in the foreseeable future..." (Article 271). It also provides a non-exhaustive list of elements to consider, such as the duration of the marriage, the age and health of the spouses, their retirement situation, and their professional situation and trajectories, as well as the impact of time devoted to the children or to the development of the other spouse's career on these trajectories ("the consequences of the professional choices made by one of the spouses during the marriage, whether to care for the children and the time that will still need to be devoted to them, or to support the career of their partner at the expense of their own," Art. 271). For this reason, numerous unofficial scales have emerged, but rather than resolving the ambiguity, they tend to increase it by often yielding divergent results [Sayn, 2018].

This study focuses in particular on the COMPRES [3] dataset, co-constructed by the Bureau for Theoretical and Applied Economics, the *Centre de Recherches Critiques sur le Droit* (Critical Legal Research Center) and the statistical department of the French Ministry of Justice. This dataset contains 5,453 judicial decisions issued by *tribunaux de grande instance* in 2013 across France, described by hundreds of variables [Jeandidier et al., 2020, Bourreau-Dubois et al., 2022, Jeandidier, 2024]. From this dataset, we use a subset of 772 divorce cases in which there was a disagreement over the CA, ultimately leading the judge to determine the amount. This dataset also includes only cases where the CA was awarded to the wife (the reverse situation represents only 4% of cases). Furthermore, we focus on 25 variables selected by domain experts (authors of the previously cited works), a few examples of which are provided below (the full list is available in Appendix B):

- `Requested amount`

- `Offered amount`

[3] https://anr.fr/Projet-ANR-12-BSH1-0002

- Age of the spouses

- Health status of the spouses

- Number of dependent children

- Difference in living standards

Existing attempts to predict the CA in the literature show that the task is challenging [Jeandidier et al., 2020, Bourreau-Dubois et al., 2022, Jeandidier, 2024]. In particular, least squares linear regression applied to 14 of the 25 selected predictors yields an absolute relative error between predicted and true CA exceeding 60% on average over the dataset [Jeandidier, 2024]. Therefore, this study aims to evaluate to what extent a capacity-based preference model, more expressive than a linear model due to its ability to capture potential interactions between variables, yet remaining interpretable and simple through the learning of a sparse Möbius capacity representation, can improve predictive performance and our understanding of the underlying decision-making mechanisms. In the following, we first describe the experimental setting and then provide the numerical results.

Data pre-processing First, the dataset is reduced to $t = 647$ examples by removing cases with missing values. Then, to represent each decision case as an evaluation vector $(x_1, \dots, x_n)$, where x_i denotes the performance with respect to criterion i, expressed on a common scale across all criteria, the variables known to have a negative impact on the CA (as identified by domain experts) are multiplied by -1 (see Table 8.4 in Appendix B). Subsequently, each column is standardized, i.e., for any $i \in N$, the transformation $x_i^\ell \leftarrow (x_i^\ell - \mu_i)/\sigma_i$, for $\ell = 1, \dots, t$, with $\mu_i = \frac{1}{t}\sum_{\ell=1}^{t} x_i^\ell$ and $\sigma_i = \sqrt{\frac{1}{t}\sum_{\ell=1}^{t}\left(x_i^\ell - \mu_i\right)^2}$ is applied.

Model specification The tests are conducted using the Choquet integral, i.e., $\Phi_S(x_S) = \min_{i \in S}\{x_i\}$, as the results obtained were better than those achieved with the multilinear utility. Furthermore, as $n = 25$ is too high to compute the vectors $\Phi(x^\ell), \ell = 1, \dots, t$ of size $2^{25} - 1 \approx 33M$, we combine our algorithm with a k-additive constraint for $k = 3$, and thus Möbius coefficients m_S with $|S| > 3$ are set to zero.

Algorithms parameters As the data consists of overall evaluation examples given by the CA set by the judges, we use the variant of D-IRLS for the regression (see Algorithm 3.3). We used $\epsilon = 1 \times 10^{-10}$ (with a maximum number of iterations equal to 150), $\eta = 1 \times 10^{-50}$, $\nu = 0$, and (δ, λ) are set by cross-validation. The quadratic programm solved at each iteration, which reduces to a support vector regression in the regression setting (see Problem 3.40), is solved using the LIBSVM library [Chang and Lin, 2011]. Finally, we compared our method to three baselines: least squares linear regression (implemented

Model	Loss	Median Error
linear	least square	$38.95\% \pm 4.96\%$
	δ-insensitive	$34.60\% \pm 4.54\%$
	ℓ_1-regularized δ-insensitive	$33.25\% \pm 2.74\%$
Choquet (D-IRLS[4])	ℓ_1-regularized δ-insensitive	$\mathbf{31.78\% \pm 2.43\%}$

Table 3.3: Generalization performance according to the model and loss function.

using the `scikit-learn` library), a variant employing the same loss function as our approach (the δ-insensitive loss), and a third variant that combines the δ-insensitive loss with ℓ_1-regularization. The latter two methods are solved via linear programming using the Gurobi solver. The parameters δ (for the loss) and λ (for the regularization) are also selected through cross-validation. This setting allows to asses the benefit of using 1) a different loss function, 2) ℓ_1 regularization and 3) an aggregation function accounting for interactions between variables.

Numerical results All algorithms are assessed according to the *median error* that is computed as the median absolute relative error between the predicted and true CA values on some test sets. The reported value corresponds to the average of this metric across 5 cross-validation folds. It is also important to note that the predicted value is systematically adjusted before the comparison with the true value: it is assimilated to the requested amount if it exceeds it, and to the offered amount if it falls below it. This is because the amount set by the judge must always lie between the offered and requested amounts [Jeandidier, 2024].

The results are provided in Table 3.3, where it can be observed that the prediction task is indeed challenging, as the errors do not fall below 30% for any of the methods. However, a significant improvement is noted between the first baseline and our method, which can be attributed to the combined effect of three components: the use of the δ-insensitive loss, the ℓ_1-regularization, and the Choquet integral.

In a second experiment, the same tests are conducted on two distinct subsets of the dataset: the cases where the CA was set above the median level (i.e., €20k), and those where it was set below. The results are presented in Tables 3.4 and 3.5, respectively. It is first observed that the prediction task is easier in the first data group, as the errors reported in Table 3.4 drop to 23%. While accounting for interactions via the Choquet integral does not appear to provide any benefit over the linear model (the identical errors

[4] Following the IRLS algorithm, an additional step is performed, consisting in solving the initial learning problem (see Problem 3.39 in Appendix B) by linear programming using the Gurobi solver, based solely on the variables selected by D-IRLS (otherwise, the performance is slightly lower).

between D-IRLS and the third baseline indicate that it returns the same, hence linear, model), the contribution of ℓ_1-regularization appears to be critical. Indeed, with only two variables (namely, the `requested amount` and `offered amounts`), the best generalization performance is achieved. Then, in Table 3.5, we observe that the errors are higher, indicating that the most difficult cases to predict are those in which the awarded CA is below the median. However, accounting for interactions leads to a clear improvement in performance, while also allowing the use of fewer variables (10 compared to 25 for the linear models).

Model	Loss	Median Error	Nb. of selected var.
	least square	$29.22\% \pm 4.92\%$	25
linear	δ-insensitive	$26.36\% \pm 5.20\%$	25
	ℓ_1-regularized δ-insensitive	$\mathbf{23.31\% \pm 4.39\%}$	**2**
Choquet	ℓ_1-regularized δ-insensitive	$\mathbf{23.31\% \pm 4.39\%}$	**2**

Table 3.4: Generalization performance on cases with CA above the median.

Model	Loss	Median Error	Nb. of selected var.
	least square	$30.13\% \pm 5.45\%$	25
linear	δ-insensitive	$32.59\% \pm 5.87\%$	25
	ℓ_1-regularized δ-insensitive	$32.63\% \pm 5.82\%$	25
Choquet	ℓ_1-regularized δ-insensitive	$\mathbf{27.21\% \pm 3.26\%}$	**10**

Table 3.5: Generalization performance on cases with CA below the median.

4 Conclusion

We have addressed the problem of preference learning with interacting viewpoints 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 Möbius representations of capacities, leading to simple models with sparse interaction patterns. This approach applies to instances possibly involving more than 20 viewpoints 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 viewpoints. 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:

- *going further in scalability*: the D-IRLS algorithm requires the solving of quadratic programs whose size depends on the number of preference examples at each iteration and therefore, does not scale when the training database exceeds a few hundred. While this may be sufficient in standard multi-criteria decision-making contexts, certain situations can involve a very large number of preference examples, such as in the case of a continuous stream of preference examples linked to user actions on social networks, search engines, etc. In order to deal with large-scale preference data, both in terms of the number of viewpoints and the number of examples, and possibly deal with incoming flows of preference examples, specific approaches are required. This direction will be investigated in Chapter 6.
- *extending the approach to learn the interaction function ϕ_S from preference data*: interaction terms occurring in the model may be more general than min, max or product of criterion values, and could also be learned from preference data. In this case, the term $m_S\phi_S(x_S)$ can be seen as a utility factor $u_S(x_S)$ in an additive decomposition of the form $\sum_{S\subseteq N} u_S(x_S)$. This leads to the problem of learning GAI-decomposable utility functions (see Subsection 1.4 of Chapter 1), which is the topic of the next chapter.

Chapter 4
Learning GAI-decomposable Utility Functions

Contents

Summary

In this chapter, we focus on the *GAI-decomposable utility function* model which allows completely general interactions between attributes while preserving some additive decomposability of the evaluation model. We present a learning approach able to identify the factors of interacting attributes and to learn the utility functions defined on these factors. This approach relies on the determination of a *sparse* representation of the *(classical or anchored) ANOVA decomposition* of the multiattribute utility function using *multiple kernel learning*. It applies to continuous and discrete attributes, and is formulated for learning from both overall evaluation and preference examples. Numerical tests are performed to demonstrate the practical efficiency of the learning approach. This chapter builds upon and extends the following publication: [Herin et al., 2024b].

Introduction

In this chapter, we address the problem of learning a general *multiattribute utility function* in the presence of *interacting attributes*, with the aim of keeping the model as simple and decomposable as possible. To this end, we focus on *GAI-decomposable utility functions* [Fishburn, 1970, Bacchus and Grove, 1995] (see Definition 1.18), which consists of a sum of interaction factors $\sum_{S \in \mathcal{F}} u_S(x_S)$ defined on a collection $\mathcal{F}$ of subsets of N (without any assumption on the kind of interactions). Such a utility model offers great flexibility in preference modeling, allowing for the capture of complex decision-making behaviors. On the other hand, maintaining a form of additive decomposability helps keep the model as simple as possible, allowing for compact preference representations, which can be exploited to derive efficient elicitation and recommendation procedures [Braziunas and Boutilier, 2008, Dubus et al., 2009, Brafman and Engel, 2010, Amor et al., 2016].

The construction of a GAI utility model from preference information (overall evaluations or pairwise comparisons) remains a challenge. It requires the determination of the relevant factors to be used in the decomposition (groups of interacting attributes) as well as the determination of sub-utility functions on these factors. Some contributions focus on the elicitation of these sub-utility functions, assuming the decomposition of the utility into factors is known [Gonzales and Perny, 2004, Braziunas and Boutilier, 2005, Braziunas, 2012]. Some of these elicitation procedures rely on a graphical representation of GAI decompositions known as *GAI-networks* [Gonzales and Perny, 2004, Gonzales et al., 2008], which closely resemble junction graphs used for Bayesian networks [Koller and Friedman, 2009]. The analogy between probability distribution decompositions (in product of marginal distributions) and utility GAI decompositions (in sum of utility factors) has been further exploited to determine the GAI decomposition using probabilistic graphical model construction algorithms [Brafman and Engel, 2010, Engel and Wellman, 2010]. Alternatively, a procedure to learn the GAI model (decomposition + utility functions) has been proposed [Bigot et al., 2012] in the case of Boolean attributes and interactions limited to subsets of bounded size (typically 2 elements). More recently, a procedure to determine a well-formed decomposition (defined in Subsection 1.1 of this chapter) of monotonic GAI models was proposed [Grabisch et al., 2022] wherein the interactions are limited to pairs of attributes. However, until now, the learning of general GAI models with no prior assumption on the size of the interacting groups of attributes is still understudied. All the above-mentioned contributions either assume that the structure of the GAI decomposition is known or that it is limited to interactions involving very few attributes. Moreover, most of them only consider the case of finite attribute domains.

Contributions and Organization of the Chapter In this chapter, we propose a more general procedure to learn a GAI utility model (decomposition + utility functions), kept as simple as possible, with no prior restriction on the size of interactions, and that applies to both continuous and discrete attribute domains. This is achieved by learning *sparse ANOVA decompositions* [Sobol', 2001, Kuo et al., 2010] of the utility function using *multiple kernel learning* [Lanckriet et al., 2004a, Gönen and Alpaydın, 2011]. More precisely, to facilitate the interpretation of the utility decomposition, we first propose to consider a class of uniquely defined functional decompositions, containing in particular the ANOVA decomposition and its anchored version (Section 1). Then, after introducing some background on kernel-based methods [Schölkopf, 2002] (Section 2.1), we present an approach to learn a GAI decomposition with multiple kernel learning from preference data either under the form of overall evaluations or pairwise comparison examples (Section 2.2). Finally, we show the benefit of the proposed approach on synthetic and real data (Section 3).

Notations As in the previous chapters, N denotes the set of attributes, i.e., $N = \{1, \ldots, n\}$, and alternatives are represented by vectors $x \in \mathcal{X} = X_1 \times \ldots \times X_n$, where X_i is the domain of the i^{th} attribute. Also, recall that the notation $S \subseteq N$ excludes the empty set by convention and for any $S \subseteq N$, the notation X_S (resp. x_S for any $x \in \mathcal{X}$) refers to the Cartesian product $\times_{j \in S} X_j$ (resp. refers to the restriction of x to its components in S). To simplify notations, for any $S \subseteq N$ and $i \in S$, $S \setminus \{i\}$ (resp. $S \setminus \{i, j\}$) is denoted by S_{-i} (resp. S_{-ij}), which is further simplified to $-i$ (resp. $-ij$) for $S = N$. Finally, for the sake of simplicity, $\mathcal{X}$ is identified to $[0, 1]^n$ in this chapter, therefore for any $S \subseteq N$, $X_S = [0, 1]^s$ with $s = |S|$. This is not restrictive since the attribute domains $X_i, i \in N$ can be numerically encoded and normalized. Note also that, for any set X, the function $\mathbb{1}_X(x)$ denotes the function : $x \to +\infty$ if $x \in X$ and 0 otherwise. The reader is also assumed to be familiar with the basic concepts of linear algebra such as a *vector space*, a *linear map*, an *inner product*, and the *norm* associated with it (otherwise, we refer to [Lang, 1987] or [Savage, 2018] (lecture notes)). Finally, we explicitly define below what is understood by an *integrable function* in this chapter.

Definition 4.1. *A continuous function $U : \mathcal{X} = [0, 1]^n \to \mathbb{R}$ is said integrable (in the sense of Lebesgue) if $\int_{\mathcal{X}} |U(x)| dx < \infty$. By convention, for any $S \subseteq N$, the integral of U w.r.t. variables in S only is denoted by $\int_{X_S} U(x) dx_S$.*

1 GAI Decomposition

In this chapter, we address the challenge of learning a GAI-decomposable utility function, i.e., a function $U : \mathcal{X} \to \mathbb{R}$ of the following form:

$$U(x) = \sum_{S \in \mathcal{F}} u_S(x_S), \text{ for any } x \in \mathcal{X} \tag{4.1}$$

where $\mathcal{F}$ is a collection of possibly overlapping subsets of N, referred to as a *decomposition* of U, and $u_S : X_S \to \mathbb{R}, S \in \mathcal{F}$ are sub-utility factors.

1.1 Non-uniqueness of the GAI Decomposition

Given a multiattribute utility function U defined on $\mathcal{X}$ there may exist multiple distinct GAI decompositions of this function. This is illustrated in the following example:

Example 4.1. *Function $U(x_1, x_2, x_3, x_4) = (x_1 - x_2)^2 + 2x_1(x_2 + x_3) + x_4$ could be seen as the sum of the three following factors: $u_{12}(x_1, x_2) = (x_1 - x_2)^2$, $u_{123}(x_1, x_2, x_3) = 2x_1(x_2 + x_3)$ and $u_4(x_4) = x_4$ or rewritten as the sum of four smaller factors, e.g., $u'_1(x_1) = x_1^2$, $u'_2(x_2) = x_2^2$, $u'_{13}(x_1, x_3) = 2x_1x_3$ and $u'_4(x_4) = x_4$. The latter decomposition is simpler because it includes factors of smaller arity that are subsets of the factors used in the former decomposition.*

The non-uniqueness of the GAI decomposition raises an issue of interpretability, as different decompositions can lead to different interpretations regarding the nature of the interactions between attributes. This is well illustrated in Example 4.1 where the second decomposition $U = u'_1 + u'_2 + u'_{13} + u'_4$, in contrast to the first decomposition $U = u_{12} + u_{123} + u_4$, indicates that there is no interaction within the group of attributes $\{1, 2, 3\}$ (nor within $\{1, 2\}$).

To further specify what would be a suitable GAI decomposition, one can resort to the notion of *well-formed decomposition* [Grabisch et al., 2022], formally defined as follows:

Definition 4.2. *[Grabisch et al., 2022] A GAI decomposition is well-formed if each term u_S appearing in the decomposition satisfies the following conditions:*

- *each variable in S is active, i.e., the derivative of u_S w.r.t. this variable is not identically 0,*
- *u_S cannot be further additively decomposed into terms involving a proper subset of variables*

Going back to Example 4.1, the decomposition $U = u_{12} + u_{123} + u_4$ is not well-formed as u_{12} and u_{123} can be additively decomposed into terms involving a proper subset of variables, yielding the second decomposition $U = u'_1 + u'_2 + u'_{13} + u'_4$, that is well-formed.

Focusing on the class of well-formed decompositions allows certain interpretations to be made. For instance, the existence of a well-formed decomposition without factor u_S for some $S \subseteq N$ suggests an absence of interaction between attributes in S. In particular, if interactions are restricted to pairwise interactions, the absence of a term $u_{i,j}$ in a well-formed decomposition is equivalent to a *2-independence* between attributes i and j (see Theorem 1 in [Grabisch et al., 2022]), i.e., for any $x_i, y_i \in X_i, x_j, y_j \in X_j, z_{-ij} \in X_{-ij}$:

$$\left((x_i, x_j, z_{-ij}), (y_i, x_j, z_{-ij})\right) \sim^* \left((x_i, y_j, z_{-ij}), (y_i, y_j, z_{-ij})\right), \tag{4.2}$$

where for any $(a, b), (c, d) \in \mathcal{X}^2$, $(a, b) \sim^* (c, d)$ reads as "the preference intensity between a and b equals the one between c and d", i.e., $U(a) - U(b) = U(c) - U(d)$. In words, a 2-independency between two attributes is characterized by the fact that changing the value of the i^{th} attribute from x_i to y_i induces the same change in utility whether the value of the j^{th} attribute equals x_j or y_j, everything other being equal.

However, well-formed decompositions are not uniquely defined. For instance, the decomposition $U = u'_1 + v'_2 + u'_{13} + u'_4$ is not the unique well-formed decomposition of the utility function given in Example 4.1, as the decomposition $U = u''_1 + u''_2 + u''_{13} + u''_4$ such that $u''_1 = \alpha x_1^2$, $u''_2 = u'_2$, $u''_{13} = (1-\alpha)x_1^2 + x_1 x_3$ and $u''_4 = u'_4$ is also a well-formed decomposition, for any $\alpha \in \mathbb{R}$. In order to avoid such utility transfers, we propose to consider a family of uniquely defined decompositions, including the *ANOVA decompositions*, which we introduce in the following Subsection.

1.2 ANOVA Decompositions

In this section, we first present the ANOVA (ANalysis Of VAriance) decomposition in its common form, which we refer to as *classical ANOVA* (as in [Griebel and Holtz, 2010]). Then, we explore a broader family of decompositions, encompassing, in particular, the classical ANOVA and the *anchored ANOVA* decomposition.

1.2.1 Classical ANOVA Decomposition

The (classical) ANOVA decomposition (also known as the *Sobol-Hoeffding decomposition*) [Hoeffding, 1948, Sobol', 2001] is a well-known functional decomposition exploited in particular in *global sensitivity analysis* [Da Veiga et al., 2021, Razavi et al., 2021], to quantify the relative importance of variables and their interactions in a model. Below, we give the ANOVA decomposition as provided in [Sobol', 2001]:

Definition 4.3. *An ANOVA decomposition of an integrable function $U : \mathcal{X} \to \mathbb{R}$ (see Definition 4.1) is a representation of U in the form:*

$$U(x) = f_\emptyset + \sum_{S \subseteq N} f_S(x_S), \quad \forall x \in \mathcal{X} \tag{4.3}$$

where for any $S \subseteq N$ and $i \in S$, f_S satisfies $\int_{X_i} f_S(x_{S_{-i}}, x_i)dx_i = 0, \forall x_{S_{-i}} \in \mathcal{X}_{S_{-i}}$.

Remark 4.1 (ANalysis Of VAriance). The name ANOVA (ANalysis Of VAriance) comes from the fact that, if $X = (X_1, \ldots, X_n)$ is a vector of random inputs, Equation 4.3 allows additively decomposing the variance of a random output $Y = U(x)$ across the different groups of inputs, thereby providing a way to quantify their influence on the uncertainty of Y. Specifically, if $X_1, \ldots, X_n$ are mutually independent and distributed according to a uniform distribution on $\mathcal{X}$, Equation 4.3 gives $U(X) = f_\emptyset + \sum_{S \subseteq N} f_S(X_S)$ (almost surely) with $\mathbb{E}[f_S(X_S)] = \int_{X_S} f_S(x_S)dx_S = 0$, $\forall S \subseteq N$. Additionnaly, if U^2 is integrable, for any $S \neq S'$, $\mathbb{E}[f_S(X_S)f_{S'}(X_{S'})] = \int_{X_{S \cup S'}} f_S(x_S)f_{S'}(x_{S'})dx_{S \cup S'} = 0$ and hence $\text{Cov}(f_S(X_S), f_{S'}(X_{S'})) = \mathbb{E}[f_S(X_S)f_{S'}(X_{S'})] - \mathbb{E}[f_S(X_S)]\mathbb{E}[f_{S'}(X_{S'})] = 0$, yielding the variance decomposition: $\text{Var}(Y) = \sum_{S \subseteq N} \text{Var}(f_S(X_S))$.

It is important to note that the ANOVA decomposition of an integrable function $U : \mathcal{X} \to \mathbb{R}$ is uniquely defined. We indeed have $f_\emptyset = \int_\mathcal{X} U(x)dx$ by integrating Equation 4.3. Then, by integrating the same equation over all variables except $x_i, i \in N$ we obtain $f_i(x_i) = \int_{X_{-i}} U(x)dx_{-i} - f_\emptyset$. Now, if we integrate Equation 4.3 on all variables except x_i and x_j, for some $i, j \in N$, we obtain $f_{ij}(x_i, x_j) = \int_{-ij} U(x)dx_{-ij} - f_i(x_i) - f_j(x_j) - f_\emptyset$. The process can be continued similarly to identify the factors of higher arity, yielding the following recursive formula:

$$f_S(x_S) = \int_{X_{\bar{S}}} U(x)dx_{\bar{S}} - \sum_{T \subset S} f_T(x_T), \text{ for any } S \subseteq N \text{ and any } x_S \in X_S \tag{4.4}$$

Equation 4.4 allows us to interpret the factor f_S as the part of U due to some interaction involving all and only the variables in S. It is computed by integrating U over variables in $\bar{S}$ and substracting the contributions already assigned to all proper subsets $T \subset S$, thereby isolating the unique effect of the interaction between the variables in S. We now illustrate the ANOVA decomposition computation on simple examples.

Example 4.2. *Let $U(x) = x_1 + x_2$, then we have:*

$$\begin{aligned} f_\emptyset &= \int_0^1 \int_0^1 (x_1 + x_2) dx_1 dx_2 = 1 \\ f_1(x_1) &= \int_0^1 (x_1 + x_2) dx_2 - f_\emptyset = x_1 - \frac{1}{2} \\ f_2(x_2) &= x_2 - \frac{1}{2} \\ f_{12}(x_1, x_2) &= x_1 + x_2 - f_1(x_1) - f_2(x_2) - f_\emptyset = 0 \end{aligned}$$

Example 4.3. *Now, if we consider the model given in Example 4.1, the same process leads to the following ANOVA decomposition:*

$$\begin{aligned} & f_\emptyset = \frac{5}{3},\ f_1(x_1) = x_1 + x_1^2 - \frac{5}{6},\ f_2(x_2) = x_2^2 - \frac{1}{3} \\ & f_3(x_3) = x_3 - \frac{1}{2},\ f_4(x_4) = x_4 - \frac{1}{2} \\ & f_{13}(x_1, x_3) = 2x_1x_3 - x_1 - x_3 - \frac{1}{3} \end{aligned}$$

where the ungiven factors are null.

For any $S \subseteq N$, let us now denote by P_S the operator that associates to any integrable function U, the function : $x_{\bar{S}} \to \int_{X_S} U(x_{\bar{S}}, x_S) dx_S$. Then, by Equation 4.4, we have that for any $S \subseteq N$ and any $x_S \in X_S$, $P_{\bar{S}}(U)(x_S) = \sum_{T \subseteq S} f_S(x_S)$, and therefore by the *Möbius* formula, $f_S(x_S)$ admits the following explicit definition:

$$f_S\,(x_S) = \sum_{T \subseteq S} (-1)^{|S|-|T|} P_{\bar{T}}(U)(x_T),\ \text{for any } x_S \in X_S \tag{4.5}$$

Operator P_S associates to any function U, a function that does not depend on the variables in S by integrating w.r.t. these variables. It is interesting to note that this operation could be performed in other ways, for instance by setting the variables in S to some reference values. For this reason, we present in the following the general result of [Kuo et al., 2010] which shows that a broad class of operators P_S can be considered, each yielding a particular decomposition defined without any ambiguity by Equation 4.5.

1.2.2 A General Decomposition Scheme

In this section, we adopt the formalism of [Kuo et al., 2010] in order to introduce a more general class of decompositions. For this, we first recall the notion of *projector*:

Definition 4.4. *Let V be a vector space. A projector is a linear application $P : V \to V$ such that $P \circ P = P$ where $\circ$ denotes the composition operator. A family of projectors*

$P_1, \ldots, P_n$ is said commuting if for any $i, j \in \{1, \ldots, n\}$,

$$P_i \circ P_j = P_j \circ P_i$$

Here, we consider a vector space V of real-valued functions $f : \mathcal{X} \to \mathbb{R}$ and commuting projectors $P_i : V \to V$, $i = 1, \ldots, n$ such that for any $i \in N$, and any $f \in V$:

$$P_i(f) \text{ does not depend on } x_i \text{, and } P_i(f) = f \text{ if } f \text{ does not depend on } x_i. \tag{4.6}$$

Note that the fact that a function f does not depend on a variable x_i is understood here as: for any $x, y \in \mathcal{X}$, $x_{-i} = y_{-i} \Rightarrow f(x) = f(y)$ (as in [Kuo et al., 2010]), and that the projectors are implicitly assumed to be well-defined on V. Finally, for any $S \subseteq N$, let P_S be the successive composition of the projections $P_i, i \in S$ (in an arbitrary order). The following theorem shows that each such family of projectors induces a unique decomposition for every function $U \in V$.

Theorem 4.1. *(Adapted from [Kuo et al., 2010]). Let $U \in V$ and let $\{P_i\}_{i=1}^n$ be commuting projectors satisfying Condition 4.6 and well-defined on V. Then, assume that:*

$$U(x) = f_\emptyset + \sum_{S \subseteq N} f_S(x_S), \quad \forall x \in \mathcal{X} \tag{4.7}$$

where for any $S \subseteq N$, f_S is a function of V depending only on the variables in S and satisfying for any $i \in S$, $P_i(f_S)(x_{S_{-i}}) = 0, \forall x_{S_{-i}} \in X_{S_{-i}}$. Then, function f_S satisfies:

$$f_S\left(x_S\right) = \sum_{T \subseteq S} (-1)^{|S|-|T|} P_{\bar{T}}(U)(x_T) = P_{\bar{S}}(U)(x_S) - \sum_{T \subset S} f_T(x_T), \quad \forall x_S \in X_S \tag{4.8}$$

where the notation $f_S(x_S)$ omits variables in $\bar{S}$, on which f_S does not depend.

By taking V as vector space of integrable function and $P_i(U)(.) = \int_{X_i} U(., x_i)dx_i$ for any $U \in V$, leading to $P_S(U)(.) = \int_{X_S} U(., x_S)dx_S$ for any $S \subseteq N$, we recover the (classical) ANOVA decomposition. Below, we explore another example.

The anchored ANOVA decomposition Another standard choice is to define P_i as the operator that freezes variable x_i by setting its value at some reference point $x^0 \in$ *dom* U, i.e., $P_i(U) = U(., x_i^0)$, yielding $P_S(U) = U(., x_S^0)$, for any $S \subseteq N$. The obtained decomposition is referred to as the *anchored ANOVA decomposition*, as the factors are anchored at the reference point [Sobol', 2003, Kuo et al., 2010, Griebel and Holtz, 2010]. This decomposition scheme is illustrated below on the functions of Example 4.2 and 4.3:

Example 4.4. *Let $U(x) = x_1 + x_2$, then its anchored ANOVA decomposition with reference point $x^0 = (0, \ldots, 0)$ is:*

$$\begin{aligned} f_\emptyset &= U(0,0) = 0 \\ f_1(x_1) &= U(x_1, 0) - f_\emptyset = x_1 \\ f_2(x_2) &= U(0, x_2) - f_\emptyset = x_2 \\ f_{12}(x_1, x_2) &= U(x_1, x_2) - f_1(x_1) - f_2(x_2) - f_\emptyset = 0 \end{aligned}$$

Example 4.5. *Now, if we consider the model given in Example 4.1 and 4.3, i.e., $U(x_1, x_2, x_3, x_4) = (x_1 - x_2)^2 + 2x_1(x_2 + x_3) + x_4$, the same process leads to the following decomposition:*

$$\begin{aligned} f_\emptyset &= U(0,0,0,0) = 0 \\ f_1 &= U(x_1,0,0,0) - f_\emptyset = x_1^2, \quad f_2 = U(0,x_2,0,0) - f_\emptyset = x_2^2 \\ f_3 &= U(0,0,x_3,0) - f_\emptyset = 0, \quad f_4 = U(0,0,0,x_4) - f_\emptyset = x_4 \\ f_{13} &= U(x_1,0,x_3,0) - f_1 - f_3 - f_\emptyset = 2x_1x_3 \end{aligned}$$

where the ungiven factors are null.

It is important to note that the anchored ANOVA decomposition coincides with already known decompositions of preference models into sum of interaction factors. For instance, we can remark that the expression of the *Choquet integral* and the *multilinear model* in terms of *Möbius* masses (see Equation 2.12 and 3.1) corresponds to the anchored ANOVA decompositions for the reference point $(0, \ldots, 0)$ of the functions C_w and ML_w respectively. In these decompositions, the factors indeed both satisfy $f_S(x_{-i}, 0) = 0$ for any $x_{-i} \in X_{-i}$, as they are respectively defined by $f_S(x_S) = m_S \min_{i \in S}\{x_i\}$ and $f_S(x_S) = m_S \prod_{i \in S} x_i$, where m_S is the *Möbius* mass attached to S.

Also, a related decomposition scheme has been already used for GAI decompositions [Fishburn, 1967, Braziunas and Boutilier, 2005, Braziunas, 2012], where the reference point x^0 appears under the name of *default* or *basic outcome*, and where it is assumed that U additively decomposes over a known collection of factors $\mathcal{F} = \{I_1, \ldots, I_M\}$, i.e., $U(x) = \sum_{j=1}^{M} u_j(x_{I_j})$. In this setting, the utility function can be decomposed as a sum of sub-utilities that, similarly to the anchored ANOVA, involve freezing U at the reference point on some attributes [Fishburn, 1967]:

$$U(x) = \sum_{j=1}^{M} (-1)^{j+1} \sum_{\substack{K \subseteq \{1,\ldots,M\} \\ |K| = j}} P_{\cup_{k \in K} \bar{I}_k}(U)(x_{\cap_{k \in K} I_k})$$

However, in our setting, the decomposition is not presupposed but shall rather be learned from preference data. To this end, we propose in the following a method for learning an ANOVA decomposition (whether classical or anchored) of the utility function using global evaluations or preference examples. The proposed approach leverages *kernel-based learning methods*, and in particular *multiple kernel learning*, which makes it possible to recover *sparse decompositions* (i.e., with few non-null factors), and thus obtain compact and interpretable representations of preferences. Importantly, the method could be easily applied to the learning of any decomposition of Theorem 4.1.

The next section is organized as follows: we first introduce kernel-based machine learning techniques in general terms (Subsection 2.1.1), then we explicit the example of the *support vector regression* (Subsection 2.1.2), and finally the extension to *multiple kernel learning* (Subsection 2.1.3). Then, in Subsection 2.2, we formulate the proposed approach for learning sparse ANOVA decompositions of utility functions, first using alternative overall evaluation examples (i.e., regression examples), and then using preference examples.

2 Sparse ANOVA Learning with Multiple Kernel Learning

2.1 Kernel-based Methods and Multiple Kernel Learning

Kernel-based methods gained attention in machine learning with the introduction of *support vector machines* [Boser et al., 1992, Vapnik, 1995] (see Subsection 2.2.2 of Chapter 3). Building upon classical results of functional analysis [Mercer, 1909, Aronszajn, 1950, Kimeldorf and Wahba, 1971], these methods led to numerous developments, including *multiple kernel learning* [Lanckriet et al., 2004a, Bach et al., 2004]. This subsection aims to provide a brief but self-contained introduction to these methods. For a more in-depth study, the interested reader may refer to the books [Schölkopf, 2002, Shawe-Taylor et al., 2004, Steinwart, 2008].

2.1.1 Kernel-based Learning Algorithms

Kernel-based learning algorithms can be approached from two different perspectives, which we detail below in the context of a regression task from a set of examples $\{(x^\ell, y^\ell)\}_{\ell=1}^t$, where $x^\ell \in \mathcal{X}$ and $y^\ell \in \mathbb{R}$.

The mapping function perspective Kernel-based learning algorithms involve embedding the input data into a high-dimensional space $\mathcal{H}$ using a (potentially) non-linear *mapping function* $\phi : \mathcal{X} \to \mathcal{H}$. The space $\mathcal{H}$, referred to as the *feature space*, may take on different forms. For instance, we could have $\mathcal{H} = \mathbb{R}^d$ with a potentially high dimension d. Below are examples of mapping functions valued in such a space, which we already encountered in Subsection 2.2.2 of Chapter 3:

- $\phi(x) = (\min_{i \in S}\{x_i\})_{S \subseteq N}$ of size $d = 2^n - 1$ (4.9)
- $\phi(x) = (\prod_{i \in S} x_i)_{S \subseteq N}$ of size $d = 2^n - 1$ (4.10)
- $\phi(x) = (x_1^2, \dots, x_n^2, \sqrt{2}x_n x_{n-1}, \dots, \sqrt{2}x_n x_1, \dots, \sqrt{2}x_2 x_1, \sqrt{2}x_n c, \dots, \sqrt{2}x_1 c, c)$ (4.11) of size $d = \dfrac{(n+2)(n+1)}{2}$

The quadratic mapping function given by Equation 4.11 could be generalized to consider polynomial features of degree up to some integer $d \in \mathbb{N}^*$. Taking a step further, ϕ could potentially include all polynomial features of degree $d \in \mathbb{N}$, embedding x in a infinite-dimensional space $\mathcal{H}$. Indeed, the only requirement on $\mathcal{H}$ is that it is a *Hilbert* space, i.e., a complete[1] vector space endowed with an inner product $\langle ., . \rangle_{\mathcal{H}}$ and the associated norm $\|.\|_{\mathcal{H}} = \sqrt{\langle ., . \rangle_{\mathcal{H}}}$

The potential non-linearity in the data being captured in ϕ, the learning task boils down to fitting a linear model in $\mathcal{H}$, i.e., finding the regression function $f(x) = \langle w, \phi(x) \rangle_{\mathcal{H}} + b$, $w \in \mathcal{H}, b \in \mathbb{R}$ that best fits the examples. Specifically, the weight vector w can be learned by minimizing both the error on the examples and the oscillation of f in $\mathcal{H}$ by solving the following ℓ_2-regularized empirical risk minimization problem:

$$\min_{w \in \mathcal{H}, b \in \mathbb{R}} C \sum_{\ell=1}^{t} l(\langle w, \phi(x^\ell) \rangle_{\mathcal{H}} + b, y^\ell) + \frac{1}{2}\|w\|_{\mathcal{H}}^2 \tag{4.12}$$

where $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a regression loss and the regularization hyperparameter $C \in \mathbb{R}_+$ allows controlling the tradeoff between fitting the examples well and minimizing oscillation to prevent overfitting (see Section 3.1.1 of Chapter 1).

Remark 4.2 (Euclidean inner product). For $\mathcal{H} = \mathbb{R}^d$, $\langle ., . \rangle_{\mathcal{H}}$ is the Euclidean inner product attached with the Euclidean norm, i.e., $\langle w, \phi(x) \rangle_{\mathcal{H}} = w^\top \phi(x)$ and $\|w\|_{\mathcal{H}}^2 = w^\top w$. Also remark that for $\phi(x) = x$, f is the linear model $f(x) = w^\top x = \sum_{i=1}^n w_i x_i$ and for ϕ given by Equation 4.9 and 4.10, we recover the Choquet integral and multilinear model (see Equation 2.12 and 3.1 respectively).

A important result is the *representer theorem*, originally due to [Kimeldorf and

[1] i.e., for any sequence $\{x_n\} \subset \mathcal{H}$ whose elements become arbitrarily close in the sense of $\|.\|_{\mathcal{H}}$ when $n \to \infty$, there exists $x \in \mathcal{H}$ such that $\lim_{n \to \infty} x_n = x$.

Wahba, 1971]. The latter states that, while $\mathcal{H}$ might be of infinite dimension, any solution of Problem 4.12 can be characterized by a finite set of variables. The proof is given to aid comprehension.

Theorem 4.2 (representer theorem). *(originally due to [Kimeldorf and Wahba, 1971]) For any solution w of Problem 4.12, there exists $(\alpha_1, \dots, \alpha_t) \in \mathbb{R}^t$ such that $w = \sum_{\ell=1}^t \alpha_\ell \phi(x^\ell)$.*

Proof. *Let E be the sub-vector space of $\mathcal{H}$ defined as the span of $(\phi(x_1), \dots, \phi(x_t))$ in $\mathcal{H}$, i.e., $E = \{\sum_{\ell=1}^t \alpha_\ell \phi(x^\ell) | (\alpha_1, \dots, \alpha_t) \in \mathbb{R}^t\}$. As E is of finite dimension, $\mathcal{H}$ can be orthogonally decomposed as follows: $\mathcal{H} = E \oplus E^\perp$, i.e., for any $w \in \mathcal{H}$, there exists $u \in E$, $v \in E^\perp$ such that $w = u + v$ ($E^\perp$ is the space of vector $v \in \mathcal{H}$ such that $\langle u, v\rangle = 0$ for any $u \in E$). Denote by J the objective function of Problem 4.12 and let $(w, b) \in \mathcal{H} \times \mathbb{R}$ be any of its solutions. Then, $w = u + v, u \in E, v \in E^\perp$ and since for any $\ell \in \{1, \dots, t\}$, $\langle v, \phi(x^\ell)\rangle_{\mathcal{H}} = 0$, $J(w, b) = J(u, b) + \frac{1}{2}\|v\|^2_{\mathcal{H}}$. Suppose by contradiction that $v \neq 0_{\mathcal{H}}$, then $\|v\|^2_{\mathcal{H}} > 0$ and $J(u, b) < J(w, b)$, which contradicts the fact that (w, b) is a solution of Problem 4.12. Therefore $v = 0_{\mathcal{H}}$ and $w \in E$.*

Therefore, by Theorem 4.2, the learned function reads as follows:

$$\begin{aligned} f(x) = \langle w, \phi(x)\rangle_{\mathcal{H}} + b &= \sum_{\ell=1}^t \alpha_\ell \langle \phi(x^\ell), \phi(x)\rangle_{\mathcal{H}} + b \\ &= \sum_{\ell=1}^t \alpha_\ell \kappa(x^\ell, x) + b \end{aligned} \tag{4.13}$$

where for any $x, x' \in \mathcal{X}$, $\kappa(x, x') = \langle \phi(x), \phi(x')\rangle_{\mathcal{H}}$ is referred to as the *kernel* function attached to ϕ. Intuitively, κ gives a measure of the similarity between x and x', once mapped in the space $\mathcal{H}$. Also, let K denotes the *kernel matrix associated with* κ and the examples, i.e., $K_{\ell\ell'} = \kappa(x^\ell, x^{\ell'})$ for any $\ell, \ell' \in \{1, \dots, t\}$. Remarking that $f(x^\ell) = (K\alpha)_\ell + b, \ell = 1, \dots, t$ and $\|w\|^2_{\mathcal{H}} = \alpha^T K \alpha$, Problem 4.12 can be reformulated as an optimization problem with finite-dimensional variables:

$$\min_{\alpha \in \mathbb{R}^t, b \in \mathbb{R}} C \sum_{\ell=1}^t l((K\alpha)_\ell + b, y^\ell) + \frac{1}{2}\alpha^T K \alpha \tag{4.14}$$

Hence, if the computation of the inner products $\kappa(x, x') = \langle \phi(x), \phi(x')\rangle_{\mathcal{H}}$ can be done efficiently, without even requiring the computation of the high-dimensional vectors $\phi(x)$, Problem 4.14 provides us with a tractable way of addressing the initial learning Problem 4.12. We saw in Chapter 3, that it is indeed the case for the mapping functions given by Equation 4.11, 4.9 and 4.10, respectively corresponding to the *quadratic*, *Choquet* and *multilinear kernel* whose formulation is recalled in Table 4.1. We also give the formulation of the *polynomial kernel*, whose mapping function is made up of polynomials of degree $d \in \mathbb{N}^*$, and of the *Gaussian kernel*, whose mapping function can be considered as an expansion of polynomials whose degree goes to infinity. Note that the polynomial kernel and its special cases involve an intercept parameter $c \in \mathbb{R}$, and the Gaussian kernel involves a variance parameter $\sigma \in \mathbb{R}_*^+$.

Kernel name	$\phi(x)$	$\kappa(x, x')$
Linear	$(x_1, \ldots, x_n, \sqrt{c})$	$x^\top x' + c$
Quadratic	$\left(\frac{\sqrt{2!}}{\sqrt{j_1! j_2! \cdots j_n! j_{n+1}!}} x_1^{j_1} x_2^{j_2} \cdots x_n^{j_n} \sqrt{c}^{j_{n+1}}\right)_{\sum_{q=1}^{n+1} j_q = 2}$	$(x^\top x' + c)^2$
Polynomial	$\left(\frac{\sqrt{d!}}{\sqrt{j_1! j_2! \cdots j_n! j_{n+1}!}} x_1^{j_1} x_2^{j_2} \cdots x_n^{j_n} \sqrt{c}^{j_{n+1}}\right)_{\sum_{q=1}^{n+1} j_q = d}$	$(x^\top x' + c)^d$
Gaussian	$e^{-\frac{\lVert x \rVert_2^2}{2\sigma^2}} \left(\left(\frac{x_1^{j_1} x_2^{j_2} \cdots x_n^{j_n}}{\sqrt{j_1! j_2! \cdots j_n! \sigma^d}}\right)_{\sum_{q=1}^{n} j_q = d}\right)_{d=0}^{\infty}$	$\exp\left(-\frac{\lVert x - x' \rVert^2}{2\sigma^2}\right)$
Choquet	$(\min_{i \in S}\{x_i\})_{S \subseteq N}$	see Prop. 3.3
Multilinear	$(\prod_{i \in S} x_i)_{S \subseteq N}$	see Prop. 3.4

Table 4.1: Examples of mapping functions and their corresponding kernels.

Remark 4.3 (universal kernel). The Gaussian kernel is a *universal kernel*, i.e., if κ is the Gaussian kernel, a function of the form $g(x) = \sum_{\ell=1}^{t} \alpha_\ell \kappa(x^\ell, x)$ can approximate arbitrary well any continuous real-valued function f defined over a compact (closed and bounded) subset of $\mathcal{X}$, provided that the number of training examples t is sufficiently high [Micchelli et al., 2006].

The kernel perspective Until here, we considered the kernel attached to a predefined mapping function ϕ. However, the starting point may be the kernel directly, meaning that we could only define a way of measuring similarities between points in $\mathcal{X}$, compute the associated kernel matrix K for the training data at hand, and solve Problem 4.14. This reveals one strength of kernel-based algorithms, as the input data could be of any nature (text, graph,...), as long as we can define similarities between inputs.

This perspective requires defining kernel functions, without considering mapping functions:

Definition 4.5. *A function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel if it is symmetric and positive definite, i.e.,:*

- *for any $x, x' \in \mathcal{X}$, $\kappa(x, x') = \kappa(x', x)$*
- *for any $\alpha_1, \ldots, \alpha_t \in \mathbb{R}^t$, $(x^1, \ldots, x^t) \in \mathcal{X}^t$, $\sum_{\ell,\ell'=1}^t \alpha_\ell \alpha_{\ell'} \kappa(x^\ell, x^{\ell'}) \geq 0$*

Remark that for any mapping function $\phi : \mathcal{X} \to \mathcal{H}$, function $\kappa(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$ satisfies the conditions of Definition 4.5. Conversely, any kernel function corresponds to an implicit mapping function, as stated by the following theorem:

Theorem 4.3. *[Aronszajn, 1950] If $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel, there exists a Hilbert space $\mathcal{H}$ and a mapping function $\phi : \mathcal{X} \to \mathcal{H}$ such that $\kappa(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$.*

The proof requires constructing from κ, an admissible mapping function ϕ and feature space $\mathcal{H}$. For this, we can exploit a specific class of Hilbert space called *reproducing kernel Hilbert space* (RKHS), which we define below:

Definition 4.6. *A reproducing kernel Hilbert space (RKHS) is an Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$, for which there exists a kernel κ such that:*

- *for any $x \in \mathcal{X}, \kappa(x, .) \in \mathcal{H}$*
- *for any $f \in \mathcal{H}, x \in \mathcal{X}, \langle f, \kappa(x, .) \rangle_{\mathcal{H}} = f(x)$ (reproducing property)*

Let κ be a kernel function. Let us now consider the associated vector space $\mathcal{H}$ of functions $f : \mathcal{X} \to \mathbb{R}$ defined by $\mathcal{H} = \{\sum_{i=1}^m \alpha_i \kappa(x^\ell, .) | \ (\alpha_1, \ldots, \alpha_m) \in \mathbb{R}^m, (x^1, \ldots, x^m) \in \mathcal{X}^m, m \in \mathbb{N}\}$. This vector space can be endowed with the inner product defined for any pairs of functions $f = \sum_{i=1}^m \alpha_i \kappa(x^i, .)$ and $g = \sum_{j=1}^n \beta_j \kappa(x^j, .)$, by $\langle f, g \rangle_{\mathcal{H}} = \sum_{i=1}^m \sum_{j=1}^n \alpha_i \beta_j \kappa(x^i, x^j)$, thus forming a Hilbert space [2]. Then we have that, for any $f \in \mathcal{H}$ and any $x \in \mathcal{X}$, $k(x, .) \in \mathcal{H}$ and that the reproducing property holds since we have:

$$\begin{aligned} \langle f, \kappa(x, .) \rangle_{\mathcal{H}} &= \langle \sum_{i=1}^m \alpha_i \kappa(x^i, .), \kappa(x, .) \rangle_{\mathcal{H}} \\ &= \sum_{i=1}^m \alpha_i \kappa(x^i, x) = f(x) \end{aligned}$$

[2] it can easily be checked that $\langle ., . \rangle_{\mathcal{H}}$ satisfies the definition of an inner product and that $\mathcal{H}$ is complete if it includes infinite sums $\sum_{i=1}^\infty \alpha_i \kappa(x^i, .)$ [Schölkopf, 2002].

Therefore $\mathcal{H}$ corresponds to the RKHS of κ. Note that while several mapping functions ϕ and feature spaces $\mathcal{H}$, can be associated to a kernel κ (i.e., such that $\kappa(x, x') = \langle\phi(x), \phi(x')\rangle_{\mathcal{H}}$), there is a one-to-one correspondance between a kernel and its RKHS [Schölkopf, 2002].

Finally, by taking the mapping function $\phi(x) = \kappa(x, .)$ and the feature space $\mathcal{H}$ as the RKHS of κ, by definition of the inner product $\langle ., .\rangle_{\mathcal{H}}$, we have that $k(x, x') = \langle\phi(x), \phi(x')\rangle_{\mathcal{H}}$, and thus we found a mapping function associated with κ. Remark that this amounts to assigning to each point $x \in \mathcal{X}$, a function giving its similarity w.r.t. all other points $x' \in \mathcal{X}$. For instance, with the Gaussian kernel we have for any $\sigma > 0$, $\phi(x) = k(x, .) = e^{-\frac{\|x-.\|^2}{2\sigma^2}}$.

Remark 4.4 (Mercer's theorem). Note that a result similar to Theorem 4.3 is given by Mercer's theorem [Mercer, 1909], the proof of which uses a different feature space $\mathcal{H}$ than the RKHS of κ and different mapping functions.

Then, solving Problem 4.14 with some kernel κ, amounts to solving Problem 4.12 with $\mathcal{H}$ as the RKHS of κ, ϕ as the mapping function : $x \to \kappa(x, .)$ and the regression function $f(x) = \langle w, \phi(x)\rangle_{\mathcal{H}} + b$. Remark that by the reproducing property (see Definition 4.6), for any $x \in \mathcal{X}$, and $w \in \mathcal{H}$, $\langle w, \phi(x)\rangle_{\mathcal{H}} = w(x)$ and therefore, the learning problem can be formulated as follows:

$$\min_{w \in \mathcal{H}(\kappa), b \in \mathbb{R}} \sum_{\ell=1}^{t} l(w(x^\ell) + b, y^\ell) + \|w\|^2_{\mathcal{H}(\kappa)} \tag{4.15}$$

where $\mathcal{H}(\kappa)$ denotes the RKHS of κ from now on. Also, it is important to note that the functions $w \in \mathcal{H}(\kappa)$ inherit most of the kernel properties such as continuity and differentiability properties [Steinwart, 2008]. In particular any function in the RKHS of the Gaussian kernel is infinitely differentiable (see exercice 4.7 in [Steinwart, 2008]).

By varying the loss function in Problem 4.14 (or equivalently in Problem 4.15), we can recover a wide range of well-known kernel-based algorithms. It is worth noting that when the loss is convex, the problem is a convex optimization problem. An emblematic example is the *support vector regression* [Smola and Schölkopf, 2004] using the ϵ-insensitive loss (introduced in the next Subsection) and its binary classification counterpart *support vector machines* (see Subsection 2.2.2 of Chapter 3) using the hinge loss (i.e., $l(f(x), \hat{y}) = \max\{0, 1 - f(x)y\}, y \in \{-1, 1\}$). We can also mention *kernel ridge regression* using the squared loss (i.e., $l(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$), *kernel logistic regression*, using the log-likelihood (classification setting) [Schölkopf, 2002] or even the *kernel canonical correlation analysis* using the correlation between two sets of variables [Akaho, 2006]. In the following, we explicit the *support vector regression* learning problem, as it serves as the basis of our learning algorithm.

2.1.2 Support Vector Regression

The *support vector regression* (SVR) is an instance of Problem 4.14 with the ϵ-*insensitive loss*, defined for any $\epsilon > 0$ as follows for any $y, \hat{y} \in \mathbb{R}$:

$$l_\epsilon(y,\hat{y}) = \begin{cases} 0 & \text{if } |y-\hat{y}| \leq \epsilon \\ |y-\hat{y}| - \epsilon & \text{otherwise} \end{cases} \tag{4.16}$$

The ϵ-insensitive loss is a convex loss that only penalizes errors that exceed a tolerance threshold ϵ, using the absolute deviation. Remark that for $\epsilon = 0$, we recover the absolute loss (i.e., $l(y,\hat{y}) = |y-\hat{y}|$). This loss can be linearized using auxiliary variables $\epsilon_\ell^+, \epsilon_\ell^- \geq 0$ modeling respectively the positive and negative part of the loss suffered on the ℓ^{th} example (see Remark 2.2 for details on linearization of absolute values). This yields the following optimization problem:

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^t,\, \epsilon^+,\epsilon^- \in \mathbb{R}^t_+, b \in \mathbb{R}} \quad & C\sum_{i=1}^{t}(\epsilon_\ell^+ + \epsilon_\ell^-) + \frac{1}{2}\alpha^\top K\alpha \\ & y^\ell - (K\alpha)_\ell - b \leq \epsilon + \epsilon_\ell^+, \quad \ell = 1,\dots,t \\ & (K\alpha)_\ell + b - y^\ell \leq \epsilon + \epsilon_\ell^-, \quad \ell = 1,\dots,t \end{aligned} \tag{4.17}$$

where ϵ^+, ϵ^- respectively denote the vector of slack variables $(\epsilon_1^+,\dots,\epsilon_t^+)$ and $(\epsilon_1^-,\dots,\epsilon_t^-)$.

Problem 4.17 is a convex quadratic optimization problem with linear constraints, and thus using Remark 3.2 (Chapter 3), it can be equivalently solved in its Lagrangian dual formulation, which admits a more compact formulation as shown hereafter. Let Y denotes the vector of output values $(y^1,\dots,y^t)$, then the Lagrangian function of Problem 4.17 can be derived by introducing Lagrange multipliers $\mu^+, \mu^-, \beta^+, \beta^- \in \mathbb{R}^t_+$ respectively attached to the example and sign constraints:

$$\begin{aligned} \mathcal{L} =& C\mathbf{1}^\top(\epsilon^+ + \epsilon^-) + \frac{1}{2}\alpha^\top K\alpha - (\epsilon^+)^\top\beta^+ - (\epsilon^-)^\top\beta^- + \Big(Y - K\alpha - b\mathbf{1} - \epsilon\mathbf{1} - \epsilon^+\Big)^\top \mu^+ \\ &+ \Big(K\alpha + b\mathbf{1} - Y - \epsilon\mathbf{1} - \epsilon^-\Big)^\top \mu^- \end{aligned}$$

where $\mathbf{1}$ denotes the vector of size t whose components are all equal to one. Then, the stationarity KKT condition (see Theorem 3.2) gives:

$$\nabla_\alpha \mathcal{L} = K\alpha - K(\mu^+ - \mu^-) = 0 \tag{4.18}$$

$$\nabla_b \mathcal{L} = -\mathbf{1}^\top(\mu^+ - \mu^-) = 0 \tag{4.19}$$

$$\nabla_{\epsilon^+} \mathcal{L} = C\mathbf{1} - \mu^+ - \beta^+ = 0, \nabla_{\epsilon^-} \mathcal{L} = C\mathbf{1} - \mu^- - \beta^- = 0 \tag{4.20}$$

A solution of Equation 4.18 is $\alpha = \mu^+ - \mu^-$. Finally, substituting these equations back into the Lagrangian gives the following dual problem:

$$\begin{aligned} \max_{\mu^+, \mu^- \in [0,C]^t} \; & -(\mu^+ - \mu^-)^\top K(\mu^+ - \mu^-) + Y^\top(\mu^+ - \mu^-) - \epsilon\mathbf{1}^\top(\mu^+ + \mu^-) \\ & \mathbf{1}^\top(\mu^+ - \mu^-) = 0 \end{aligned} \tag{4.21}$$

If μ^+, μ^- are optimal solutions of Problem 4.21, using $\alpha = \mu^+ - \mu^-$, we obtain the following learned regression function:

$$f(x) = \sum_{\ell=1}^{t} (\mu_\ell^+ - \mu_\ell^-)\kappa(x^\ell, x) + b \tag{4.22}$$

where b can be recovered by accessing the dual optimal variables of the unique constraint of Problem 4.21

Remark 4.5 (kernel trick). Problem 4.21 can be equivalently recovered from the initial learning problem (see Problem 4.12) without resorting to the representer theorem (see Theorem 4.2), but simply by applying Lagragian duality. More precisely, let ϕ be any mapping function valued in $\mathcal{H} = \mathbb{R}^d$, then Problem 4.12 for the ϵ-insensitive loss is the following problem (after linearization):

$$\begin{aligned} \min_{w, \epsilon^+ \in \mathbb{R}_+^t, \epsilon^- \in \mathbb{R}_+^t} \; & C\sum_{\ell=1}^{t}(\epsilon_\ell^+ + \epsilon_\ell^-) + \frac{1}{2}\|w\|_2^2 \\ & y^\ell - w^\top\phi(x^\ell) - b \leq \epsilon + \epsilon_\ell^+, \quad \ell = 1, \ldots, t \\ & w^\top\phi(x^\ell) + b - y^\ell \leq \epsilon + \epsilon_\ell^-, \quad \ell = 1, \ldots, t \end{aligned}$$

Then, if μ^+, μ^- denote the dual variables of the constraints in the problem given above, it can easily be checked that its dual coincides with Problem 4.21, where K is the matrix such that $K_{\ell,\ell'} = \phi(x^\ell)^\top\phi(x^{\ell'})$. The fact that Problem 4.21 only depends on these inner products supports the idea that one could consider the kernel matrix associated with any kernel function $\kappa(x, x') = \langle\phi(x), \phi(x')\rangle_{\mathcal{H}}$, even if $\mathcal{H}$ has infinite dimension. This observation is known as the *kernel trick* [Schölkopf, 2002] and is also frequently used to approach kernel-based methods independently of the representer theorem, as was done in Subsection 2.2.2 of Chapter 3 for support vector machines.

Problem 4.21 is a convex quadratic program that can be solved using standard numerical solvers implementing interior-point methods (see Subsection 3.1.4 of Chapter 1). Furthermore, efficient iterative procedures that take advantage of the specific structure of the optimization problem have been proposed. A major example is the *sequential minimal optimization* (SMO) algorithm and its variants [Platt, 1998, Fan et al., 2005] that is efficiently implemented in the library LIBSVM [Chang and Lin, 2011].

In the following, we return to Problem 4.14 for an arbitrary loss function, and introduce its extension to multiple kernel learning.

2.1.3 Multiple Kernel Learning

The core idea of Multiple Kernel Learning (MKL) [Lanckriet et al., 2004a, Bach et al., 2004] is to use a basis of kernels $\{\kappa_l\}_{l=1}^p$ and a non-negative weight vector $d = (d_1, \dots, d_p)$ to replace κ with a linear combination of kernels $\boldsymbol{\kappa}_d = \sum_{l=1}^p d_l \kappa_l$. Note that $\boldsymbol{\kappa}_d$ is still a kernel as the non-negativity of the weights preserves the positive semi-definite property (see Definition 4.5). Also, in the following, the kernel matrix associated with $\boldsymbol{\kappa}_d$ is denoted by $\boldsymbol{K}_d$, and we have $\boldsymbol{K}_d = \sum_{l=1}^p d_l K_l$ where K_l denotes the kernel matrix associated with κ_l.

MKL may be used to achieve two different types of objectives [Gönen and Alpaydın, 2011]:

(1) to combine the different similarity measures induced by the different kernels to uncover sophisticated data patterns and/or to automatically select the kernels that work best on the training data by learning a sparse weight vector d, thus avoiding the bias that would come from pre-selecting a specific kernel. For instance, it is used in computer vision [Varma and Ray, 2007, Bucak et al., 2013, Gu et al., 2017] where different kernels are used to accounts for similarities related to different image characteristics such as color, shape or texture.

(2) to combine different sources of information, when each kernel κ_l takes as input a different group of variables. This is of interest for assigning different kernels to feature groups of different nature that may require distinct similarity measures, or for *performing a selection of the important variable groups via a sparse learning of d* (with or without distinct kernels for each group). For instance, it is used in genomics to combine data of different nature such as DNA sequences, gene expression and protein expression [Lanckriet et al., 2004b, Yu et al., 2013, Wilson et al., 2019].

In both cases, introducing this weighted combination into a kernel-based learning methods requires the learning of both the model parameters α and the weight vector d. This challenge can be handled in multiple ways (an overview of the different methods is provided in [Gönen and Alpaydın, 2011]). Here, we focus on optimization approaches that search for d that minimizes both the optimum value of Problem 4.14 obtained with $K = \boldsymbol{K}_d$, denoted by $J_{l,C}(\boldsymbol{K}_d)$ for a loss $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and a parameter $C \in \mathbb{R}_+$, and a regularization term $\Omega(d)$, as follows:

$$\min_{d \in \mathbb{R}_+^p} J_{l,C}(\boldsymbol{K}_d) + \lambda \Omega(d) \tag{4.23}$$

where $\lambda \in \mathbb{R}_+$ is an hyperparameter controlling the level of regularization on d.

Whatever it is for selecting kernels (setting (1)) or selecting feature groups (setting (2)), $\Omega(d)$ is often taken as a convex sparsity-inducing regularization, such as ℓ_1-regularization either in the objective (i.e., $\Omega(d) = \|d\|_1$) [Varma and Ray, 2007], or in constraint (i.e., $\Omega(d) = \mathbb{1}_{\mathcal{B}_1}(d)$ where $\mathcal{B}_1 = \{d \in \mathbb{R}^p | \|d\|_1 \leq 1\}$) [Bach et al., 2004, Zien and Ong, 2007, Rakotomamonjy et al., 2007, 2008], which are known to be equivalent (see Theorem 1 in [Kloft et al., 2011]). As d is non-negative, its ℓ_1-norm reduces to $\|d\|_1 = \sum_l d_l$, and the constrained version of the regularization often appears as an equality, i.e., $\sum_l d_l = 1$ [Bach et al., 2004, Zien and Ong, 2007, Rakotomamonjy et al., 2007, 2008]. We can also mention approaches using a trace constraint on $\boldsymbol{K}_d$ [Lanckriet et al., 2004a, Qiu and Lane, 2008], which reduces to a weighted version of ℓ_1-regularization by linearity of the trace (i.e., $\Omega(d) = \mathbb{1}_{\mathcal{T}_c}(d)$ where $c > 0$ and $\mathcal{T}_c(d) = \{d \in \mathbb{R}^p | \mathrm{tr}(\boldsymbol{K}_d) = \sum_{l=1}^p d_l \mathrm{tr}(K_l) \leq c\}$). Finally, other approaches use egularizations that do not induce sparsity such as ℓ_2-regularization [Cortes et al., 2009] and more generally ℓ_p-regularizations [Kloft et al., 2009, 2011] (i.e., $\Omega(d) = \|d\|_p^p$, $p \geq 1$; see Definition **??**), or entropy regularization (i.e., $\Omega(d) = \sum_{l=1}^p d_l \ln(d_l)$) [Xu et al., 2010].

It is important to note that, *if l and Ω are convex functions, Problem 4.23 is a convex optimization problem.* This can be proved using a reasoning similar to that of [Bach et al., 2012] (Subsection 1.5). First, remark that for any loss l and $C > 0$, $J_{l,C}(\boldsymbol{K}_d)$ can be reformulated as the optimum of a constrained problem:

$$J_{l,C}(\boldsymbol{K}_d) = \min_{\alpha, u \in \mathbb{R}^t, b \in \mathbb{R}} R_{l,C}(u, b) + \frac{1}{2} \sum_{l=1}^p d_l \alpha^T K_l \alpha \quad \text{s.t.} \qquad u = \sum_{l=1}^p d_l K_l \alpha \tag{4.24}$$

where $R_{l,C}(u, b) = C \sum_{\ell=1}^t l(u_\ell + b, y^\ell)$. The Lagrangian function of Problem 4.24 then reads as $\mathcal{L} = R_{l,C}(u, b) + \frac{1}{2} \sum_{l=1}^p d_l \alpha^T K_l \alpha + \gamma^\top (u - \sum_{l=1}^p d_l K_l \alpha)$, where $\gamma \in \mathbb{R}^t$ are the dual variables. Also, if l is convex, then $R_{l,C}$ is convex and Problem 4.24 is a convex optimization problem with equality constraints. Therefore, by Remark 3.2 of Chapter 3, strong duality holds and Problem 4.24 may be equivalently written in its dual form:

$$\begin{aligned} J_{l,C}(\boldsymbol{K}_d) &= \max_{\gamma\in\mathbb{R}^t} \min_{\alpha\in\mathbb{R}^t, b\in\mathbb{R}} R_{l,C}(u,b) + \frac{1}{2}\sum_{l=1}^{p} d_l \alpha^T K_l \alpha + \gamma^\top(u - \sum_{l=1}^{p} d_l K_l \alpha) \\ &= \max_{\gamma\in\mathbb{R}^t}(\min_{u\in\mathbb{R}^t, b\in\mathbb{R}} R_{l,C}(u,b) + \gamma^\top u) + (\min_{\alpha\in\mathbb{R}^t} \frac{1}{2}\sum_{l=1}^{p} d_l \alpha^T K_l \alpha - \gamma^\top \sum_{l=1}^{p} d_l K_l \alpha) \\ &= \max_{\gamma\in\mathbb{R}^t} \left((\min_{u\in\mathbb{R}^t, b\in\mathbb{R}} R_{l,C}(u,b) + \gamma^\top u) - \frac{1}{2}\sum_{l=1}^{p} d_l \gamma^T K_l \gamma \right) \end{aligned} \tag{4.25}$$

Hence, $J_{l,C}(\boldsymbol{K}_d)$ is the maximum of a function linear in d, and is thus convex in d. Indeed, it can easily be checked that any pointwise maximum of a linear function, i.e., function of the form $h(\alpha) = \max_x x^\top \alpha$ is convex.

When l is the ϵ-insensitive loss or the hinge loss (i.e., $J_{l,C}(\boldsymbol{K}_d)$ is the optimum of a SVR or SVM problem), Problem 4.23 often reduces to a convex *quadratically constrained quadratic program* (QCQP) (for instance see [Lanckriet et al., 2004a, Qiu and Lane, 2008] for a trace regularization, and [Rakotomamonjy et al., 2008] for a ℓ_1-norm constraint regularization). Such optimization problem can be solved for a reasonable number (hundreds) of examples and kernels using standard numerical solvers implementing interior-point methods (see Subsection 3.1.4 of Chapter 1). To consider problems of larger size, more efficient iterative optimization procedures have been proposed [Sonnenburg et al., 2006, Varma and Ray, 2007, Rakotomamonjy et al., 2008, Kloft et al., 2011].

In the following, we exploit the idea of selecting group features with MKL (setting (2)) to learn sparse GAI-decomposition of utility functions. While kernel-based methods have been widely used to learn utility functions from preference examples [Chapelle and Harchaoui, 2004, Radlinski and Joachims, 2005, Waegeman et al., 2009, Lahaie, 2010, Domshlak and Joachims, 2012, Tehrani et al., 2014b, Tehrani, 2021], to the best of our knowledge, there is no attempt on learning GAI decompositions of these utility functions exploiting additively decomposed kernels. For this reason, we propose in the following a method for learning a sparse (classical or anchored) ANOVA decomposition of the utility function using overall evaluations or preference examples.

2.2 Sparse ANOVA Learning from Regression and Preference Examples

It is important to note that there is a large body of work on estimating, from regression example, a functional decomposition of the form of Equation 4.7 (often referred to as *high-dimensional model representations* (HDMR) [Rabitz and Aliş, 1999, Li et al., 2001, Sobol', 2001]). A popular method for the classical ANOVA is *smoothing splines ANOVA* (SS-ANOVA) [Wahba, 1990, Wahba et al., 1995, Gu, 2002, 2014], and its extension to learn sparse decomposition *COmponent Selection and Smoothing Operator* (COSSO) [Lin and Zhang, 2006]; both of them can be considered as kernel-based learning methods using specific losses and kernels aligned with the classical definition of ANOVA decomposition. However, in this chapter, we take the general view of multiple kernel learning that is a flexible framework accommodating a wide range of loss functions and kernel decompositions, and that enables the learning of sparse decompositions via convex optimization.

This section is organized as follows: we first discuss the choice of a kernel basis well-suited to learn a (classical or anchored) ANOVA decomposition of the utility function (Subsection 2.2.1). Then we explicit a QCQP formulation of Problem 4.23 with $J_{l,C}(\boldsymbol{K}_d)$ as the optimum of a SVR problem and $\Omega(d)$ as an ℓ_1-regularization term, to obtain sparse decompositions of the utility functions from examples of overall evaluations of alternatives $\{(x^\ell, y^\ell)\}_{\ell=1}^t$, $y^\ell \in \mathbb{R}$ (Subsection 2.2.3). Finally, we extend the proposed method for learning the utility decomposition from preference examples $\{x^\ell, x'^\ell\}_{\ell=1}^t$ where $x^\ell \succ x'^\ell$ for any $\ell \in \{1, \dots, t\}$, using a basis of *preference kernel* (Subsection 2.2.4).

2.2.1 All-subsets Kernel Basis

Let us consider a basis of kernel functions $\{\kappa_S\}_{S \subseteq N}$ where for any $S \subseteq N$, κ_S is a kernel depending on, and only on, the attributes in S. Such a basis, referred to as an *all-subsets kernel basis* in the following, can be constructed using a univariate kernel $k : X_1 \times X_1 \to \mathbb{R}$ and considering the tensor product of k, i.e., for any $x_S, x'_S \in \mathcal{X}_S$:

$$\kappa_S(x_S, x'_S) = \prod_{i \in S} k(x_i, x'_i) \tag{4.26}$$

Note that the tensor product of kernels is a kernel, as the product of kernels is a kernel (see Proposition 13.2 in [Schölkopf, 2002]). Then, we consider the kernel combination:

$$\boldsymbol{\kappa}_d = \sum_{S \subseteq N} d_S \kappa_S \tag{4.27}$$

where d is now indexed by $S \subseteq N$ (in the lexicographical order). Remark that, by solving Problem 4.14 using $\boldsymbol{\kappa}_d$, we learn a function $u \in \mathcal{H}(\boldsymbol{\kappa}_d)$ decomposed into a sum of factors:

$$U(x) = \sum_{\ell=1}^{t} \alpha_\ell \boldsymbol{\kappa}_d(x^\ell, x) + b = \sum_{S \subseteq N} d_S \sum_{\ell=1}^{t} \alpha_\ell \kappa_S(x_S^\ell, x_S) + b = \sum_{S \subseteq N} u_S(x_S) + b \tag{4.28}$$

with $u_S(x_S) = d_S \sum_{\ell=1}^{t} \alpha_\ell \kappa_S(x_S^\ell, x_S) \in \mathcal{H}(\kappa_S)$, for any $S \subseteq N$.

For $d = (1, \ldots, 1)$, $\boldsymbol{\kappa}_d = \sum_S \kappa_S(x_S, x'_S) = \sum_S \prod_{i \in S} k(x_i, x'_i) = \prod_{i=1}^{n}(1 + k(x_i, x'_i))$ is a tensor product kernel that is well-known under the name *ANOVA kernel* [Vapnik, 1998, Saunders et al., 1998, Stitson et al., 1999]. However, if the name is related to the ANOVA's core idea of decomposing a function in terms depending on subsets of variables, this kernel does not provide an ANOVA decomposition since nothing guarantees that $\int_{X_i} u_S(x_{S_{-i}}, x_i) dx_i = 0$ or $u_S(x_{S_{-i}}, 0_i) = 0$, $\forall x_{S_{-i}} \in \mathcal{X}_{S_{-i}}$, for any $S \subseteq N$ and $i \in S$, which are the respective conditions defining the classical and anchored ANOVA decompositions (see resp. Definition 4.3 for the classical and Subsection 1.2.2 for the anchored version). Also, the ANOVA kernel uses all the possible subsets of N (or all until a given size p) and as it is, does not allow to perform a selection of the most important subsets.

In the following subsection, exploiting Theorem 4.1, we explicit a condition on the univariate kernel κ so that the decomposition $\{u_S\}_{S \subseteq N}$ given by Equation 4.28 coincides with the (classical or anchored) ANOVA decomposition of the learned function U. Then in Subsection 2.2.3, we address the challenge of learning a sparse decomposition using MKL, and in particular by formulating the learning problem as an instance of Problem 4.23 with ℓ_1-regularization.

2.2.2 Retrieving an ANOVA Decomposition

Let us consider one of the two following conditions on the univariate kernel k:

$$\int_{X_1} k(x, x') dx' = 0, \text{ for any } x \in X_1 \tag{4.29}$$

$$k(x, 0) = 0, \text{ for any } x \in X_1 \tag{4.30}$$

Then, we have the following implications:

$$(4.29) \Rightarrow \int_{X_i} u_S(x_{S_{-i}}, x_i) dx_i = d_S \sum_{\ell=1}^{t} \alpha_\ell \prod_{j \neq i} k(x_j^\ell, x_j) \int_{X_i} k(x_i^\ell, x_i) dx_i = 0, \forall S, i \in S, x_{S_{-i}} \in X_{S_{-i}}$$

$$(4.30) \Rightarrow u_S(x_{S_{-i}}, 0_i) = d_S \sum_{\ell=1}^{t} \alpha_\ell \prod_{j \neq i} k(x_j^\ell, x_j) k(x_i^\ell, 0) = 0, \quad \forall S, i \in S, x_{S_{-i}} \in X_{S_{-i}}$$

Therefore, if k satisfies Condition 4.29 (resp. Condition 4.30), the decomposition $\{u_S\}_{S \subseteq N}$ coincides with the classical (resp. anchored) ANOVA decomposition of the learned function U. Some standard univariate kernel verify Condition 4.29 or 4.30 by definition. For Condition 4.29, a well-known example is the *Sobolev kernel* of order r used in SS-ANOVA [Wahba, 1990, Gu, 2013], constructed with Bernouilli polynomials as follows:

$$k(x,x') = \frac{B_{2r}(|x-x'|)}{(-1)^{r+1}(2r)!} + \sum_{i=1}^{r} \frac{B_i(x)B_i(x')}{(i!)^2}. \tag{4.31}$$

where for any $r \in \mathbb{N}$, B_r is the Bernouilli polynomial of degree r, which satisfies $\int_0^1 B_r(x)dx = 0$ (see [Abramowitz and Stegun, 1968] Chapter 23). However, one may be interested in identifying the function U in other RKHS, associated with different regularity properties and approximation capabilities. This can be done by constructing an univariate kernel $k^0(x,x')$ from any univariate kernel k, the integral of which (w.r.t x or x') equals zero, as proposed in [Durrande et al., 2013]:

$$k^0(x,x') = k(x,x') - \frac{\int_s k(s,x')ds \int_t k(x,t)dt}{\int_s \int_t k(s,t)dsdt} \tag{4.32}$$

Under mild hypothesis on kernel k, it can be shown that the attached k^0 is still a kernel [Durrande et al., 2013], and it is straightforward to see that k^0 satisfies Condition 4.29 by construction. Note that k is required to verify $\int k(s,x)ds < \infty$ for any $x \in \mathcal{X}$ and $\int\int k(s,t)dsdt < \infty$ so that k^0 is well defined.

Below, we also give examples of standard kernels of varying flexibility verifying Condition 4.30:

- $k(x,x') = xx'$ (*linear univariate kernel*) (4.33)
- $k(x,x') = (xx')^d$ (*polynomial univariate kernel* of degree $d \in \mathbb{N}$ with $c = 0$) (4.34)
- $k(x,x') = \min(x,x')$ (*brownian kernel* [Karatzas and Shreve, 1991]) (4.35)
- $k(x,x') = xx' + \frac{(x+x')\min(x,x')}{2} - \frac{\big(\min(x,x')\big)^3}{6}$ (*first order infinite spline kernel* [Vapnik et al., 1996]) (4.36)

In the following, the univariate kernel k is assumed to satisfy either Condition 4.29 or Condition 4.30, depending on whether a classical ANOVA decomposition or an anchored decomposition is desired.

2.2.3 Sparse ANOVA Decomposition

In order to select the most useful coalitions and provide simple ANOVA decompositions, we need to learn a sparse representation of the weight vector d. To this end, we combine the SVR algorithm with ℓ_1-regularization to obtain sparsity in d, as it is done in [Gunn and Kandola, 2002] with SVR and ridge regression and in [Varma and Ray, 2007] with SVM. This amounts solving the MKL problem (see Problem 4.23) with $J_{l,C}(\boldsymbol{K}_d)$

as the optimum value of the SVR Problem (see Problem 4.21 in the dual or 4.17 in the primal) and $\Omega(d) = \|d\|_1$.

Note that [Gunn and Kandola, 2002] also considers an all-subset kernel basis to learn ANOVA decompositions, making their initial problem equivalent to ours. In particular, the univariate kernel k is set to the first order infinite spline kernel (see Equation 4.36), which allows for uncovering anchored ANOVA decompositions. However, the possibility of learning different types of ANOVA decompositions (classical or anchored) is not discussed and the optimization task is tackled using alternative minimization w.r.t. α and d, whereas we show in the following that the learning problem can be formulated as a compact QCQP, which, as we show in Section 3, can be solved in reasonable time using standard numerical solvers up to a dozen attributes.

Finally, [Durrande, 2011] (Chapter 5) also proposes to learn sparse classical ANOVA decompositions using MKL. However, they exploit the *hierarchical multiple kernel* approach [Bach, 2008], that consists in using a group ℓ_1-regularization that would automatically include a factor u_S with its sub factors $u_{S'}, S' \subseteq S$. Such mechanism allows obtaining efficient optimization algorithms with polynomial time complexity in the number of selected kernels [Bach, 2008], however, may not always be desirable. For instance, coming back to the function $U(x_1, x_2, x_3, x_4) = (x_1 - x_2)^2 + 2x_1(x_2 + x_3) + x_4 = x_1^2 + x_2^2 - x_1x_3 + x_4$ used in Example 4.1, 4.3 and 4.5, the anchored ANOVA decomposition (see Example 4.5) is given by:

$$u_1(x_1) = x_1^2, u_2(x_2) = x_2^2, u_4(x_4) = x_4, u_{13}(x_1, x_3) = x_1x_3$$

all the other factors being null. Therefore, the factor u_{13} is included, while its sub-factor u_3 is not. Therefore, such a decomposition could not be recovered with hierarchical regularization.

Learning Problem and QCQP Dual Formulation Let us consider a set of regression examples $\{(x^\ell, y^\ell)\}_{\ell=1}^t$ and denote by $K_S, S \subseteq N$ the kernel matrices associated with the basis $\{\kappa_S\}_{S\subseteq N}$. We solve an instance of Problem 4.23 with $\boldsymbol{K}_d = \sum_{S\subseteq N} d_S K_S$, $J_{l,C}(\boldsymbol{K}_d)$ as the optimum value of the SVR Problem and ℓ_1-regularization. Integrating the SVR problem using its dual formulation (see Problem 4.21), and recalling that since the weights d_S are positive, the ℓ_1-penalty is simply the sum of the weights, we have the following optimization problem:

$$\min_{d \in \mathbb{R}_+^{2^n-1}} \max_{\substack{\mu^+,\mu^- \in [0,C]^t \\ \mathbf{1}^\top(\mu^+-\mu^-)=0}} \Big(-\sum_{S \subseteq N} d_S (\mu^+ - \mu^-)^\top K_S (\mu^+ - \mu^-) + Y^\top(\mu^+ - \mu^-) - \epsilon \mathbf{1}^\top (\mu^+ + \mu^-)\Big) + \lambda \sum_{S \subseteq N} d_S \tag{4.37}$$

which can be reformulated as:

$$\min_{d \in \mathbb{R}_+^{2^n-1}} \max_{\substack{\mu^+,\mu^- \in [0,C]^t \\ \mathbf{1}^\top(\mu^+-\mu^-)=0}} \Big(\sum_{S \subseteq N} d_S \big(\lambda - (\mu^+ - \mu^-)^\top K_S (\mu^+ - \mu^-)\big) + Y^\top(\mu^+ - \mu^-) - \epsilon \mathbf{1}^\top (\mu^+ + \mu^-)\Big)$$

Therefore, Problem 4.37 corresponds to the Lagrangian dual of the following optimization problem, where only the quadratic constraints have been dualized and d_S are their corresponding dual variables:

$$\begin{aligned} \max_{\mu^+,\mu^- \in [0,C]^t} \; & Y^\top(\mu^+ - \mu^-) - \epsilon \mathbf{1}^\top (\mu^+ + \mu^-) \\ & \lambda - \frac{1}{2}(\mu^+ - \mu^-)^\top K_S (\mu^+ - \mu^-) \geq 0, \quad S \subseteq N \\ & (\mu^+ - \mu^-)^\top \mathbf{1} = 0 \end{aligned} \tag{4.38}$$

Remark 4.6. Lagrangian duality applies to maximization problem by remarking that $\max_{g(x) \leq 0} F(x) \iff -\min_{g_i(x) \leq 0, i=1,\dots,m} -F(x)$. Then by using the definition of the Lagrangian dual of a minimization problem (see Subsection 2.2.1 of Chapter 3), we have that its dual is $\min_{d \in \mathbb{R}_+^m} \max_x F(x) - \sum_{i=1}^m d_i g_i(x)$.

As Problem 4.38 is a convex problem where the quadratic constraints admit the strictly feasible point $\mu^+ = \mu^- = \frac{C}{2}\mathbf{1}$ for any $\lambda > 0$, it satisfies Slater's condition (see Remark 3.2 of Chapter 3) and strong duality holds. Therefore Problem 4.38 and 4.37 are equivalent. Problem 4.38 is a quadratically constrained program involving $2t$ variables and 2^n constraints (where t is the number of examples and n is the number of attributes). We will show in Section 3 that it can be solved in reasonable times up to a dozen of attributes and hundreds of examples using standard numerical solvers implementing interior-point methods (see Subsection 3.1.4 of Chapter 1). Note that similar dual formulations of Problem 4.23 with $J_{l,C}(\boldsymbol{K}_d)$ corresponding to a SVR or SVM optimum can be found in the literature. However, their formulations differ slightly due to variations in the choice of regularization [Lanckriet et al., 2004a, Qiu and Lane, 2008, Rakotomamonjy et al., 2007, 2008].

If μ^+, μ^- are solutions of Problem 4.38, then similarly to the SVR (see Subsection

2.1.2), $\alpha = \mu^+ - \mu^-$, and b can be recovered by accessing the dual variable of the equality constraint. Then, the learned function reads as:

$$U(x) = \sum_{\ell=1}^{t} \alpha_\ell \boldsymbol{\kappa}_d(x^\ell, x) + b = \sum_{S \subseteq N} d_S \sum_{\ell=1}^{t} (\mu^+ - \mu^-)_\ell \kappa_S(x_S^\ell, x_S) + b = \sum_{S \subseteq N} d_S \tilde{u}_S(x_S) + b$$

where $\tilde{u}_S(x_S) = \sum_{\ell=1}^{t} (\mu^+ - \mu^-)_\ell \kappa_S(x_S^\ell, x_S)$. It is interesting to note that, by the complementary slackness KKT condition (see Theorem 3.2), we have that $d_S\Big((\mu^+ - \mu^-)^\top K_S(\mu^+ - \mu^-) - \lambda\Big) = 0$ for any $S \subseteq N$, and therefore as soon as $(\mu^+ - \mu^-)^\top K_S(\mu^+ - \mu^-) \Leftrightarrow \|\tilde{u}_S\|^2_{\mathcal{H}(\kappa_S)} < \lambda$, we have $d_S = 0$. As the presence of a sub-utility u_S in the decomposition of the utility function U is equivalent to a non-null weight d_S, we thus recover simpler decompositions by increasing the ℓ_1-penalty hyper-parameter λ in Problem 4.38.

In the following, we extend the proposed method to cope with learning examples under the form of pairwise comparisons.

2.2.4 Learning from Pairwise Preference Examples

In this section, we learn a utility function from pairwise comparison examples of the form $\{(x^\ell, x'^\ell)\}_{\ell=1}^t$ where $x^\ell \succ x'^\ell$ for any $\ell \in \{1, \dots, t\}$, and seek the utility function that best captures the decision maker's preference ranking .

Preference Kernel Similarly to the kernel-based regression setting, the utility function is modeled as $U(x) = \langle w, \phi(x) \rangle_{\mathcal{H}}$, where $\mathcal{H}$ is a Hilbert space and $\phi : \mathcal{X} \to \mathcal{H}$ is a mapping function that may include non-linearities. Note that the intercept term is omitted in this context, as it is not necessary for explaining preference examples.

Preference example violations can be penalized using the convex pref-hinge loss $l(U(x), U(x')) = \max\{0, \delta - (U(x') - U(x))\}$ (see Definition 1.28) where $\delta \geq 0$ is a tolerance threshold. As this loss only depends on the utility difference $U(x') - U(x) = \langle w, \phi(x) - \phi(x') \rangle_{\mathcal{H}}$, the learning problem can be formulated as follows:

$$\min_{w \in \mathcal{H}} C \sum_{\ell=1}^{t} g(\langle w, \tilde{\phi}(x^\ell, x'^\ell) \rangle_{\mathcal{H}}) + \frac{1}{2}\|w\|^2_{\mathcal{H}} \tag{4.39}$$

where $g(s) = \max\{0, \delta - s\}$ for any $s \in \mathbb{R}$ and $\tilde{\phi} : \mathcal{X} \times \mathcal{X} \to \mathcal{H}$ is the mapping function such that $\tilde{\phi}(x, x') = \phi(x) - \phi(x')$. The proof of the representer theorem (see Theorem 4.2) can be easily adapted to Problem 4.39 and we obtain that for any solution w, there exists $\alpha \in \mathbb{R}^t$ such that $w = \sum_{\ell=1}^t \alpha_\ell \tilde{\phi}(x^\ell, x'^\ell)$. Then, the learned function is:

$$\begin{aligned} U(x) = \langle w, \phi(x) \rangle_{\mathcal{H}} &= \langle \sum_{\ell=1}^t \alpha_l \tilde{\phi}(x^\ell, x'^\ell), \phi(x) \rangle_{\mathcal{H}} \\ &= \sum_{\ell=1}^t \alpha_l (\langle \phi(x^\ell), \phi(x) \rangle_{\mathcal{H}} - \langle \phi(x'^\ell), \phi(x) \rangle_{\mathcal{H}}) = \sum_{\ell=1}^t \alpha_l (k(x^\ell, x) - k(x'^\ell, x)) \end{aligned} \tag{4.40}$$

where κ is the kernel function associated with ϕ. Intuitively, the utility of an alternative $x \in \mathcal{X}$ is determined by its similarity (in the sense of κ) to the preferred alternatives (i.e., $\{x^\ell\}_{\ell=1}^t$) and its dissimilarity to the non-preferred ones (i.e., $\{x'^\ell\}_{\ell=1}^t$).

Let us now denote $\tilde{\kappa}$ the kernel function associated with $\tilde{\phi}$, which coincides with the *preference kernel* introduced in Subsection ? of Chapter 3, defined as follows:

$$\begin{aligned} \tilde{\kappa}((x, x'), (z, z')) &= \langle \tilde{\phi}(x, x'), \tilde{\phi}(z, z') \rangle_{\mathcal{H}} \\ &= \langle \phi(x), \phi(z) \rangle_{\mathcal{H}} + \langle \phi(x'), \phi(z') \rangle_{\mathcal{H}} - \langle \phi(x), \phi(z') \rangle_{\mathcal{H}} - \langle \phi(x'), \phi(z) \rangle_{\mathcal{H}} \\ &= \kappa(x, z) + \kappa(x', z') - \kappa(x', z) - \kappa(z', x) \end{aligned} \tag{4.41}$$

Finally, let $\tilde{K}$ denotes the kernel matrix associated with $\tilde{\kappa}$, i.e., for any $\ell, \ell' \in \{1, \dots, t\}$, $\tilde{K}_{\ell\ell'} = \tilde{\kappa}((x^\ell, x'^\ell), (x^{\ell'}, x'^{\ell'}))$. Then, similarly to the regression setting, Problem 4.39 reformulates as an optimization problem with a finite number of variables:

$$\min_{\alpha \in \mathbb{R}^t} C \sum_{\ell=1}^t g((\tilde{K}\alpha)_\ell) + \frac{1}{2} \alpha^T \tilde{K} \alpha \tag{4.42}$$

Similarly to the SVR Problem (see Subsection 2.1.2), Problem 4.42 can be linearized by introducing positive slack variables ϵ_ℓ^+ modeling the error suffered on the ℓ^{th} example. This gives the following learning problem:

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^t, \epsilon^+ \in \mathbb{R}_+^t} \ & C \sum_{\ell=1}^t \epsilon_\ell^+ + \frac{1}{2} \alpha^\top \tilde{K} \alpha \\ & (\tilde{K}\alpha)_\ell \leq \delta + \epsilon_\ell^+, \quad \ell = 1, \dots, t \end{aligned} \tag{4.43}$$

Remark 4.7 (ranking SVM). It is worth mentioning that Problem 4.43 corresponds to a support vector machine (SVM) problem (see Subsection 2.2.2 of Chapter 3) with kernel $\tilde{K}$, margin δ, no intercept, and only positive examples. Such a problem can be encountered in the literature under the name of *ranking SVM* [Herbrich et al., 2000, Radlinski and Joachims, 2005, Evgeniou et al., 2005, Chen et al., 2009].

Finally, similarly to the SVM/SVR problem, Problem 4.43 can be equivalently solved in its dual formulation, which reduces to the following problem:

$$\max_{\mu\in[0,C]^t} -\mu^\top \tilde{K}\mu + \delta\mathbf{1}^\top\mu \tag{4.44}$$

where it can easily be checked the stationnarity KKT conditions yield $\alpha = \mu$.

Sparse ANOVA Decomposition with Mutiple Kernel Learning As in the regression setting, we can learn a decomposition of U in a sum of factors by replacing κ with a decomposed kernel $\boldsymbol{\kappa_d}$ given by Equation 4.27, yielding the following utility model:

$$U(x) = \sum_{S\subseteq N} d_S \sum_{\ell=1}^{t} \alpha_\ell(\kappa_S(x_S^\ell, x_S) - \kappa_S(x_S'^\ell, x_S)) \tag{4.45}$$

It is straightforward to see that if the univariate kernel k used to construct the kernels $\kappa_S, S\subseteq N$ satisfies Condition 4.29 (resp. Condition 4.30), we have $\int_{X_i} u_S(x_{S_{-i}}, x_i)dx_i = 0, \quad \forall S, i\in S, x_{S_{-i}}\in X_{S_{-i}}$ (resp. $u_S(x_{S_{-i}}, 0_i) = 0, \quad \forall S, i\in S, x_{S_{-i}}\in X_{S_{-i}}$) and thus, the decomposition given by Equation 4.45 coincides with the classical (resp. anchored) ANOVA decomposition of U.

Let us now denote $\tilde{\boldsymbol{K}}_d$ and $\tilde{K}_S, S\subseteq N$ the kernel matrices associated with the kernels $\tilde{\boldsymbol{\kappa}}_d$ and $\tilde{\kappa}_S$, respectively corresponding to the preference version of $\boldsymbol{\kappa}_d$ and κ_S, as defined by Equation 4.41. Naturally, we have $\tilde{\boldsymbol{K}}_d = \sum_{S\subseteq N} d_S\tilde{K}_S$. Then, a sparse (classical or anchored) ANOVA decomposition can be learned with MKL by solving an instance of Problem 4.23 where $J_{l,C}(\tilde{\boldsymbol{K}}_d)$ is the optimum value of Problem 4.44 with $\tilde{K} = \tilde{\boldsymbol{K}}_d$ and $\Omega(d)$ is an ℓ_1-regularization term, i.e.,:

$$\min_{d\in\mathbb{R}_+^{2^n-1}} \max_{\mu\in[0,C]^t} \Big(-\sum_{S\subseteq N} d_S\mu^\top\tilde{K}_S\mu + \delta\mathbf{1}^\top\mu\Big) + \lambda\sum_{S\subseteq N} d_S \tag{4.46}$$

Problem 4.46 coincides with the dual of the following quadratically constrained convex optimization problem:

$$\begin{aligned}&\max_{\mu\in[0,C]^t} \delta\mathbf{1}^\top\mu \\ &\lambda - \frac{1}{2}\mu^\top\tilde{K}_S\mu \geq 0, S\subseteq N\end{aligned} \tag{4.47}$$

which satisfies Slater's conditions and thus is equivalent to Problem 4.46. Therefore, the MKL learning task can be solved with Problem 4.47, where the optimal weights $d_S, S\subseteq N$ are recovered by assessing the dual optimal variables of the quadratic constraints.

3 Numerical Tests

This section presents the results of numerical tests performed on synthetic and real-world preference data. We implement our method, called SMKGAI for Sparse Multiple Kernel GAI, with the univariate first order infinite spline kernel (see Equation 4.36) to learn anchored ANOVA decompositions, and the univariate Gaussian kernel, i.e., $k(x, x') = \exp\big(-\frac{(x-x')^2}{2\sigma^2}\big)$ (with $\sigma = 1$) transformed according to Equation 4.32, to learn classical ANOVA decompositions. Note that in practice, the integrals involved in Equation 4.32 have to be approximated with numerical integration. To avoid unnecessary repeated computations, we use a discretized representation of function $x \mapsto \int k(s, x)ds$ that has been computed beforehand.

The tests are conducted in the regression setting, by solving Problem 4.38 with the tolerance threshold ϵ set to 0.01 and the regularization hyper-parameters C and λ selected by cross-validation using a number of folds equal to 3. 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).

3.1 Synthetic Data

We first show the result of the learning on synthetic data generated with a 6-dimensional utility function involving two irrelevant variables, i.e., $U(x_1, x_2, x_3, x_4, x_5, x_6) = x_1^2 + x_4^2 + 2x_3x_4$. We generate a regression training set of size $t = 70$ from the hidden utility function U, with a random uniform draw of alternatives x^ℓ in $[0, 1]^n$. The data is then perturbated with a centered Gaussian noise with standard error $\sigma = 0.05$. Then, an anchored ANOVA decomposition is learned using the first-order infinite spline kernel, and its significant factors are represented in red in Figure 4.1, alongside the true anchored ANOVA decomposition of U shown in black.

Secondly, we conduct an experiment using a model with a high degree of interaction: $U(x) = \sum_{i=1}^n x_i + 1000 \prod_{i=1}^n x_i$ for $n = 6$. In order to assess the benefit of allowing high interactions in the learning of a GAI decomposition, we compare SMKGAI with p-additive GAI utilities that do not use ℓ_1-regularization to select the most useful factors but that include factors of size at most p for $p \in \{1, 2, 3, 4\}$. The case $p = 1$ corresponds to the learning of an additive utility. This is done using the SVR algorithm, i.e., by solving Problem 4.21 using the ANOVA kernel of degree p: $\kappa = \sum_{S \subseteq N, |S| \leq p} \kappa_S$. The experiment is conducted to learn both anchored and classical ANOVA decompositions of U. For this, we generate, from the hidden utility function U, random regression training sets of size $t = 70$ with a random uniform draw of alternatives x^ℓ in $[0, 1]^n$ for the anchored and $[-0.5, 0.5]^n$ for the classical ANOVA decomposition (so that both ground truth decompositions of U

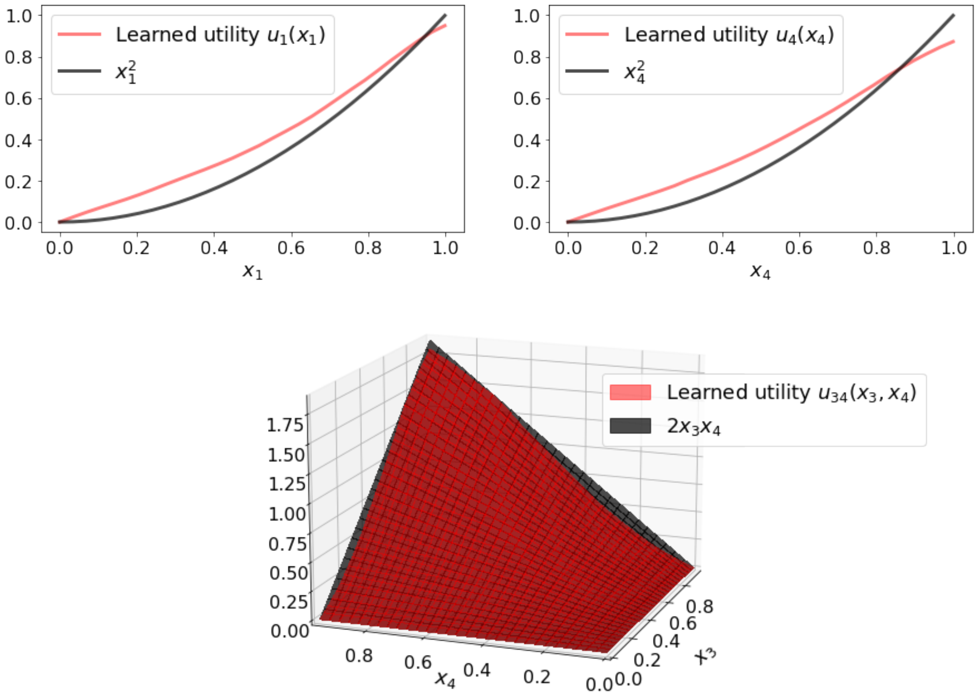


Figure 4.1: Learned and ground truth utility factors u_1(top left), u_2 (top right), u_{34} (bottom).

are $u_i(x_i) = x_i$ and $u_N(x) = 1000 \prod_{i=1}^{n} x_i)$. The data is then perturbated with a centered Gaussian noise with standard error corresponding to 2% of the standard deviation of $U(x)$.

In Table 4.2, we compare the generalizing performances of SMKGAI and the one obtained with dense p-additive GAI models (p-GAI) over 20 simulations performed with the first order infinite spline kernel (yielding anchored ANOVA decompositions). The generalized performances are measured as the relative mean absolute errors (MAE (%)), i.e., the average relative differences between the ground truth utility and the predicted utility over test sets of size 150. We also provide the computing times (sec.) for all the methods, along with the degree of interaction of the learned function, i.e., the size of the largest included coalition (max size), and the number of factors. We observe that SMKGAI, by capturing the interaction of size n, divides by 3 the MAE (%) compared to the 1-additive models, and by 2 compared to the 4-additive model, with a computation time lower than 10 seconds in average. Similar results are obtained in Table 4.3 for the learning of classical ANOVA decompositions with the Gaussian kernel transformed according to Equation 4.32.

Finally, we perform an experiment on synthetic data generated with more general models for $n = 10$. The models are randomly generated as sums of 10 tensor products of quadratic splines. In order to increase the complexity of the hidden models, the maximal

	MAE (%)	Time (sec.)	Max Size	Number of Factors
SMKGAI	$\mathbf{33.38 \pm 11.21}$	9.66 ± 0.99	6.00 ± 0.00	6.00 ± 2.45
1-GAI	104.95 ± 36.29	$\mathbf{1.62 \pm 0.22}$	1.00 ± 0.00	6.00 ± 0.00
2-GAI	73.60 ± 9.59	2.25 ± 0.28	2.00 ± 0.00	21.00 ± 0.00
3-GAI	70.73 ± 14.49	2.95 ± 0.43	3.00 ± 0.00	41.00 ± 0.00
4-GAI	67.75 ± 14.97	3.59 ± 0.45	4.00 ± 0.00	56.00 ± 0.00

Table 4.2: Comparison of SMKGAI and p-GAI (anchored ANOVA).

	MAE (%)	Time (sec.)	Max Size	Number of Factors
SMKGAI	$\mathbf{31.61 \pm 28.02}$	8.80 ± 0.64	6.00 ± 0.00	7.05 ± 0.22
1-GAI	152.70 ± 148.35	$\mathbf{1.61 \pm 0.19}$	1.00 ± 0.00	6.00 ± 0.00
2-GAI	177.15 ± 181.71	2.22 ± 0.29	2.00 ± 0.00	21.00 ± 0.00
3-GAI	159.47 ± 133.52	2.99 ± 0.37	3.00 ± 0.00	41.00 ± 0.00
4-GAI	160.80 ± 136.76	3.53 ± 0.36	4.00 ± 0.00	56.00 ± 0.00

Table 4.3: Comparison of SMKGAI and p-GAI (classical ANOVA).

size of the factors (max. size) is increased from 1 (additive utility) to 5. We perform 20 simulations and each time, we generate a set of regression examples of size $t = 140$ perturbated with a Gaussian centered noise of standard error $\sigma = 0.05$. In Table 4.4 is represented the MAE (%) on test sets of size 150 along with the maximal size of the learned factors and the False Discovery Rate (FDR), which is computed as the percentage of selected factors in the learned ANOVA decomposition that are not included in any of the factors of the hidden function. We consider that a factor $S \subseteq N$ is selected as soon as the attached weight d_S is higher than 0.01. As expected, we observe that the MAE (%) increases as the interaction degree (max size) of the hidden model increases. However, our learning approach is able to capture these interactions since the maximal size of the learned factors increases similarly to the ground truth, with a percentage of false inclusion (FDR) in the model that stays below 20%.

3.2 Real-world Datasets

In this Subsection, we test our method on real preference datasets. We use standard multi-criteria decision-making benchmarks containing overall evaluations of alternatives described by continuous or discrete attributes. We use *Employee Selection* (ESL) which contains profiles and overall psychological evaluations of job candidates, *Lecture Evaluation* (LEV), containing examples of anonymous lecturer evaluations and *Employee*

Max size (true)	MAE (%)	Max size	FDR
1	0.026 ± 0.012	1.0 ± 0.0	0.0 ± 0.0
2	0.036 ± 0.012	2.0 ± 0.4	0.011 ± 0.033
3	0.067 ± 0.013	3.0 ± 1.4	0.078 ± 0.115
4	0.085 ± 0.020	4.1 ± 1.0	0.198 ± 0.165
5	0.082 ± 0.015	4.2 ± 1.2	0.156 ± 0.124

Table 4.4: Model recovery assessment for growing interaction degree of the hidden models in average over 20 simulations.

Rejection/Acceptance (ERA)[3], which contains the judgment of a decision-maker w.r.t candidate profiles. Then from the UCI repository, we use CPU and Car MPG (MPG) which respectively contain the performances of CPU and the fuel consumption of cars, along with attributes describing the objects. Finally, we use the *Movehub city ranking*[4] (CITY) dataset which contains overall evaluations of cities quality. The number of evaluations t and the number of attributes n of each dataset is given in Table 4.5.

Dataset	ESL	LEU	ERA	CPU	MPG	CITY
n	4	4	4	6	7	5
t	488	1000	1000	209	392	216

Table 4.5: Datasets' number of attributes n and examples t.

We compare SMKGAI to standard baselines from preference modeling such as the linear regression (LR), the 2-additive Choquet Integral (2-add CI) (see Definition 1.9 and 1.10), and p-additive GAI (p-GAI) for $p = 1$ and $p = 2$. The attribute values are normalized using a linear max-min normalization. Each dataset is split to produce a training set containing 80% of the examples and a test set with the 20% left. For 20 random splits, we compute the MAE (%) obtained on the test set for each method and present the averaged results in Table 4.6 for SMKGAI and p-GAI with the Gaussian kernel (yielding classical ANOVA decompositions) and in Table 4.7 for the first order infinite spline kernel (yielding anchored ANOVA decompositions). For each dataset, the best result is displayed in bold and if there is another performance close to this result, it is also displayed in bold. We can see that SMKGAI is attached to the best average MAE (%) or is very close to the optimal result, except on the dataset ESL where the linear regression and the additive utility (1-GAI) provide the best results. In particular, for the datasets LEV and ERA, SMKGAI outperforms the baseline methods, showing

[3] www.openml.org (ESL, LEV and ERA)

[4] www.kaggle.com/datasets/blitzr/movehub-city-rankings

the presence of interactions between more than two attributes in the data. Also, for the datasets CPU, MPG and CITY, it seems that SMKGAI is able to adapt its complexity to the underlying data since it provides results similar to the additive utility (1-GAI) or 2-additive GAI (2-GAI) depending on the case.

Data	SMKGAI	1-GAI	2-GAI	LR	2-add CI
ESL	0.084 ± 0.007	$\mathbf{0.083 \pm 0.006}$	0.084 ± 0.007	$\mathbf{0.082 \pm 0.006}$	0.085 ± 0.005
LEV	$\mathbf{0.124 \pm 0.027}$	0.232 ± 0.013	0.171 ± 0.012	0.235 ± 0.006	0.254 ± 0.014
ERA	$\mathbf{0.047 \pm 0.002}$	0.201 ± 0.011	0.118 ± 0.009	0.243 ± 0.005	0.243 ± 0.007
CPU	$\mathbf{0.008 \pm 0.005}$	0.009 ± 0.002	$\mathbf{0.008 \pm 0.005}$	0.028 ± 0.004	0.018 ± 0.007
MPG	$\mathbf{0.052 \pm 0.007}$	0.056 ± 0.011	$\mathbf{0.054 \pm 0.007}$	0.064 ± 0.003	0.101 ± 0.006
CITY	$\mathbf{0.051 \pm 0.009}$	$\mathbf{0.049 \pm 0.009}$	0.051 ± 0.009	0.063 ± 0.009	0.067 ± 0.008

Table 4.6: MAE (%) averaged over 10 random splits for SMKGAI (classical ANOVA) and baseline methods.

Data	SMKGAI	1-GAI	2-GAI	LR	2-add CI
ESL	0.082 ± 0.007	0.081 ± 0.007	0.083 ± 0.007	$\mathbf{0.080 \pm 0.006}$	0.084 ± 0.007
LEV	$\mathbf{0.145 \pm 0.012}$	0.226 ± 0.013	0.212 ± 0.015	0.236 ± 0.013	0.251 ± 0.017
ERA	$\mathbf{0.036 \pm 0.007}$	0.216 ± 0.007	0.172 ± 0.007	0.252 ± 0.004	0.257 ± 0.005
CITY	$\mathbf{0.045 \pm 0.007}$	0.048 ± 0.006	0.048 ± 0.006	0.056 ± 0.006	0.066 ± 0.005

Table 4.7: MAE (%) averaged over 10 random splits for SMKGAI (anchored ANOVA) and baseline methods.

4 Conclusion

We have presented a multiple kernel learning approach that constructs a sparse GAI model from overall evaluation (regression) or pairwise comparisons examples to describe and explain the value system of a decision maker. The core of the approach relies on the determination of a sparse (classical or anchored) ANOVA decomposition of utilities obtained thanks to the use of well-suited kernels (of zero integrals or zero values at an anchor point). The advantage of the proposed approach is to be able to capture general interactions among continuous or discrete attributes without prior restrictions on the size of interacting factors. It makes it possible to fit model complexity to the available preference information. The regularization used in the objective function ensures that model complexity is kept as low as possible, given the descriptive constraints imposed by preference data. As far as we know, this is the first learning method able to learn both the structure of the GAI decomposition (by identification of the factors that really matter), and the utility functions defined on these factors, that can handle continuous attributes and that does not use prior restrictions on the cardinality of the interactions. In order to go further, some directions are worth investigating.

(1) A direction is to enhance the scalability of the method w.r.t. the number of attributes, since the number of possible factors, and therefore the number of constraints in the optimization problem to be solved (see Problem 4.38 or 4.47), grows exponentially. One path could be to bound from above the size of possible interacting factors. Another possible path is to use an efficient iterative optimization procedure to solve the MKL problem (see Problem 4.23) in the primal, for instance using gradient descent on d [Varma and Ray, 2007, Rakotomamonjy et al., 2007, 2008], or using alternating minimization on d and α [Sonnenburg et al., 2006, Kloft et al., 2011], or again using stochastic gradient descent [Orabona et al., 2012].

(2) Another interesting direction is the learning of *monotonic GAI decomposition*, i.e., such that for any $S \subseteq N$, u_S is non-decreasing w.r.t. any variable $x_i, i \in S$. Indeed, in a multicriteria decision-making setting where the elements of X_i are assumed to be ordered according to a weak order $\succsim_i$, such decomposition allows for a clear interpretation as a positive effect of a factor u_S can not be canceled by a negative effect of some other sub-factors $u'_S, S \subseteq S'$. A learning method was proposed in [Grabisch et al., 2022] wherein the interactions are limited to pairs of attributes and discrete attribute domains. Thus, an interesting direction could be to explore how the learning of monotone GAI decompositions can be extended to continuous attribute domains and arbitrary interaction degrees using multiple kernel learning. However, how monotonicity can be incorporated into our method remains unclear.

Chapter 5
Noise-tolerant Active Preference Learning for Multicriteria Choice Problems

Contents

Summary

In this chapter, we propose an *active* preference learning method for determining the weights of an aggregation function used by a DM to choose among a set of alternatives described by multiple criteria. Here, *active* means that the proposed algorithm iteratively selects the alternatives for the DM to compare instead of using a pre-collected database of preference examples, resulting in an interactive process with her. The approach not only reduces the weights indeterminacy to identify an optimal or near-optimal alternative but also learns a predictive model capable of making relevant choices for new instances of choice problems. Furthermore, the proposed approach is *noise-tolerant* in the sense that it allows for the identification of the weights that best represent the DM's preferences, even if she sometimes deviates from it in the answers. This is made possible by leveraging a general *disagreement-based active learning* approach for binary classification that is guaranteed to be tolerant to noisy answers. The proposed method applies to various weighted aggregation functions, linear or not, classically used in decision theory. This chapter is based on the following publication: [Herin et al., 2024a].

Introduction

In *multicriteria choice problems*, it is commonly accepted that the exploration of admissible trade-offs should be restricted to Pareto-optimal solutions, i.e., solutions that cannot be improved on one criterion without having to be degraded on another. These solutions are, however, potentially very numerous, and it is necessary to collect additional preference information to define how the evaluations from the different criteria combine to define the overall preference. If the DM's preferences are modeled by a utility function F_w, parameterized by some weight vector w, the multicriteria choice problem can then be reformulated as a problem of maximizing the scalarizing function over all feasible performance vectors, i.e., the evaluation vectors associated with the alternatives in the problem under consideration. We consider here a wide range of such functions from the simplest such as the weighted sum, to more sophisticated and expressive models such as the *multilinear model* and the *Choquet integral* (see Definition 1.15 and 1.9), which can be used to model interactions between criteria, without forgetting weighted norms, such as the *weighted Chebyshev Norm* (see Definition 1.16).

The phase of eliciting preferences and learning the weighting vector w is absolutely crucial, as it completely determines the nature of the compromise that will be found by optimizing F_w and the recommendation that will follow. A first family of approach named *incremental preference elicitation*, consists in progressively reducing the space of admissible parameters. Iteratively, a preference query is chosen, the answer to which induces a new constraint on the parameter space [White et al., 1984]. A principle of active question selection is often used, based on the minimization of maximum regret, to choose the most informative question [Wang and Boutilier, 2003, Boutilier et al., 2006, Benabbou et al., 2017a, 2020] and derive a robust recommendation. This principle of progressive reduction of the uncertainty attached to w reveals quite efficient in practive but implictly assumes that answers to preference queries are free of errors. Another approach, more tolerant to noisy responses, is to manage a probability distribution (or other uncertainty model [Adam and Destercke, 2024]) over the parameter space and revise it according to the answers to questions, to choose a decision having the maximum expected value [Chajewska et al., 2000] or minimizing the expectation of regret [Bourdache et al., 2019a]. Overall, incremental elicitation methods based on maximum regret are question-saving, as they direct the questionnaire towards the resolution of a particular instance. On the other hand, they do not produce a learned model and are generally not sufficient to solve a choice problem involving a new set of alternatives.

An alternative approach to the problem is to adopt the preference learning perspective (or regression-based elicitation; see Section 3.2.2 or 2.2.2 in Chapter 1) and use a dataset of preference examples to perform regression, either on the values themselves or on the order they induce, with the goal of determining the parameter w that best fits these data. To determine the parameters of the aggregation function accurately and reliably, the model must be trained on a large dataset of examples and requires significantly more preference queries than incremental approaches. On the other hand, this approach is inherently tolerant to errors in the preference examples and the learned model can be reused to solve new choice problems involving the same decision-maker and new alternatives.

Contributions and Organization of the Chapter In this chapter, we propose a *hybrid* active learning approach that combines the benefits of incremental preference elicitation and preference learning. Specifically, it enables the rapid identification of the optimal choice for the instance to be solved, while remaining resilient to noise in the DM's answers and providing a model capable of explaining the DM's preferences and predicting her choices on new instances of choice problems. To this end, we first present the principle of incremental preference elicitation and discuss its limitations in handling noisy responses (Section 1). We then present the *disagreement-based active learning* principle (Section 2.1) and propose an algorithm for active preference learning in a noisy setting (Section 2.2). Finally, in Section 3, we demonstrate its benefits using synthetic preference data.

Notations In the sequel, $N = \{1, \dots, n\}$ denotes a set of criteria, and we assume that the alternatives in the decision problem are described by their evaluation w.r.t. the n criteria, i.e., by vectors of the form $x = (x_1, \dots, x_n) \in \mathcal{X} = [0, 1]^n$. Additionally, by convention, for any $t \in \mathbb{R}$, $\text{sign}(t) = 1$ if $t \geq 0$ and $\text{sign}(t) = -1$ otherwise. Moreover, $\mathbb{1}[C]$ equals 1 when condition C is met and 0 otherwise.

1 Incremental Preference Elicitation

In this chapter, we consider a multicriteria choice problem over a finite set $X \subseteq \mathcal{X}$ representing all feasible evaluation vectors. Then, the preferences of the DM over the elements of X are modeled by a utility function in the form of an aggregation function F_w, where the set of admissible parameter values for w is denoted by $\mathcal{W}$. For instance, if F_w is a weighted sum, then $\mathcal{W}$ corresponds to the n-dimensional simplex, i.e., $\{w \in \mathbb{R}^n_+ | w_1 + \dots + w_n = 1\}$. Similarly, if F_w is the Choquet integral, $\mathcal{W}$ is the set of capacities defined on the power set of N. For a detailed introduction to standard aggregation functions, we refer the reader to Section 1.3.1. In this setting, *incremental preference elicitation* [White et al., 1984, Wang and Boutilier, 2003, Boutilier et al., 2006, Benabbou et al., 2017a, Bourdache et al., 2019a, Adam and Destercke, 2024] aims to incrementally

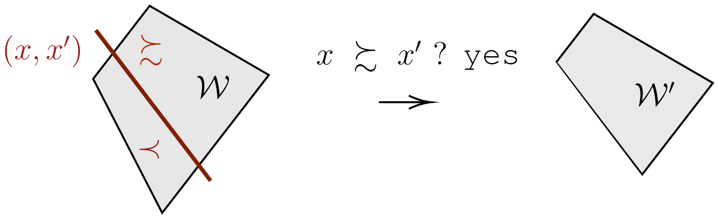


Figure 5.1: Illustration of incremental preference elicitation.

reduce the space of admissible values for parameter w, in collaboration with the DM, in order to identify its preferred solution among X.

To this end, the DM is typically asked to compare pairs of alternatives, i.e., for a pair $x, x' \in X$, to answer the question: *is x at least as good as x'?*, which we denote by $x \succsim x'$? in the following. The answer to such query can then be used to reduce $\mathcal{W}$. For instance, if the answer is yes, $\mathcal{W}$ can be reduced to $\mathcal{W}' = \{w \in \mathcal{W} | F_w(x) \geq F_w(x')\}$, as illustrated in Figure 5.1. It is important to note that if F_w is linear in its parameter w, the constraints $F_w(x) \geq F_w(x')$ induces a linear constraint on $\mathcal{W}$, justifying the representation of $\mathcal{W}$ and $\mathcal{W}'$ as polyhedrons in Figure 5.1. The process can then be continued to incrementally reduce the space $\mathcal{W}$ until it becomes sufficiently restricted to eliminate any ambiguity about the DM's preferences over X, and in particular, her most preferred option.

An active query selection strategy called the *current solution strategy* (CSS) [Wang and Boutilier, 2003] is often used to guide the elicitation process. This querying strategy is based on the notion of *pairwise maximum regret*, denoted by PMR, and defined for any pair $x, x' \in X$ and set $\mathcal{W}$ as follows:

$$\mathrm{PMR}(x, x', \mathcal{W}) = \max_{\omega \in \mathcal{W}} \{F_\omega(x') - F_\omega(x)\} \tag{5.1}$$

Intuitively, $\mathrm{PMR}(x, x', \mathcal{W})$ quantifies the worst loss of utility (over all admissible model parameter values) that the DM may suffer if x is recommended to her instead of x'. Then, for an alternative $x \in X$, we can define the worst PMR w.r.t. all alternatives $x' \in X$, using the *maximum regret*, denoted MR, and defined as follows:

$$\mathrm{MR}(x, X, \mathcal{W}) = \max_{x' \in X} \mathrm{PMR}(x, x', \mathcal{W}) \tag{5.2}$$

Then, for a given admissible set $\mathcal{W}$, a reasonable recommendation is that of the alternative in X that has the lowest maximum regret. This lowest level of regret, denoted by mMR, is referred to as the *minmax regret*, and defined by:

$$\mathrm{mMR}(X, \mathcal{W}) = \min_{x \in X} \mathrm{MR}(x, X, \mathcal{W}) \tag{5.3}$$

Indeed, let x^* be the alternative of minimal MR, i.e., $x^* \in \arg\min_{x \in X} \mathrm{MR}(x, X, \mathcal{W})$ and ϵ the associated MR, i.e., $\epsilon = \mathrm{mMR}(X, \mathcal{W})$, then we have that for any $x' \in X$ and $w \in \mathcal{W}$:

$$F_w(x^*) \geq F_w(x') - \epsilon$$

Then, x^* emerges as a (quasi)-necessary optimal alternatives. However, if ϵ is too high, additional information has to be collected to reduce the indetermination on the optimal alternative. Then, the CSS strategy consists in asking the DM to compare x^* with its best opponent i.e., $x' = \arg\max_{x' \in X} \mathrm{PMR}(x^*, x', \mathcal{W})$. Finally, the answer to the query can be used to reduce $\mathcal{W}$. This reduction necessarily yields a reduction of the mMR as for any $\mathcal{W}' \subseteq \mathcal{W}$, it can easily be checked that:

$$\mathrm{mMR}(X, \mathcal{W}) \geq \mathrm{mMR}(X, \mathcal{W}') \geq 0$$

Therefore, this process can be incrementally conducted until the mMR reaches a sufficiently low level. Such procedure is summarized in Algorithm 5.1 that takes as inputs X, an initial set of admissible parameter values $\mathcal{W}_0$ and a desired reduction percentage $\rho \in [0, 1]$ of the mMR (compared to the first iteration).

Algorithm 5.1: Incremental preference elicitation (IPE)

```
Inputs: $X$, $\mathcal{W}_0$, $\rho$
$\mathrm{mMR}_0 \leftarrow \mathrm{mMR}(X, \mathcal{W}_0)$
$\hat{x}_0 \leftarrow \arg\min_{x \in X} \mathrm{MR}(x, X, \mathcal{W}_0)$
$k \leftarrow 1$
while $\mathrm{mMR}_{k-1} > \rho \mathrm{mMR}_0$ do
    $x'^k \leftarrow \arg\max_{x' \in \mathcal{X}} \mathrm{PMR}(\hat{x}_{k-1}, x', \mathcal{W})$
    if the answer to the query $\hat{x}_{k-1} \succsim x'^k$? is yes then
        $\mathcal{W}_k \leftarrow \{w \in \mathcal{W}_{k-1} | F_w(\hat{x}_{k-1}) \geq F_w(x')\}$
    else
        $\mathcal{W}_k \leftarrow \{w \in \mathcal{W}_{k-1} | F_w(\hat{x}_{k-1}) < F_w(x'^k)\}$
    $\mathrm{mMR}_k, \hat{x}_k \leftarrow \mathrm{mMR}(X, \mathcal{W}_k), \arg\min_{x \in X} \mathrm{MR}(x, X, \mathcal{W}_k)$ ;
    $k \leftarrow k + 1$ ;
Outputs: $\hat{x}_{k-1}$
```

This approach, which aims for questionnaire efficiency, is of course rather risky, as it omits any validation operation through partial redundancy of questions, nor any compromise between partially contradictory answers within the framework of a given decision model. The pitfalls of incremental elicitation by progressive and definitive reduction of possible parameters are well illustrated by the following example.

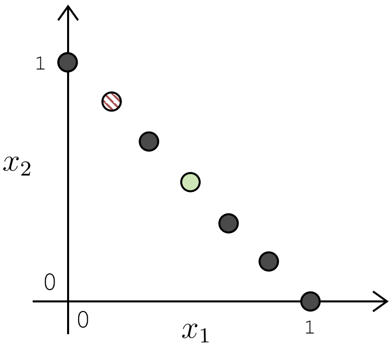


Figure 5.2: Illustration of the set of alternatives of Example 5.1.

Example 5.1. *Consider a set $X = \{a^0, \ldots, a^q\}$ of $q+1$ alternatives evaluated on two criteria and represented by performance vectors $a^i = (i/q, (q-i)/q)$ for $i = 0, \ldots, q$. Suppose the DM has expressed a first preference $a^r \succ a^t$ for two indices $r, t \in \{0, \ldots, q\}$ such that $r > t$, as illustrated in Figure 5.2 for $q = 7$, $r = 3$ and $t = 1$ (a^r and a^t are respectively represented in green and dashed red). Suppose also we want to learn the weights of a weighted sum model of the form $F_w(x) = wx_1 + (1-w)x_2$ for an unknown parameter $w \in [0,1]$. The preference $a^r \succ a^t$ implies $wa_1^r + (1-w)a_2^r > wa_1^t + (1-w)a_2^t$ and therefore $wr + (1-w)(q-r) > wt + (1-w)(q-t)$, or equivalently $w(r-t) > (1-w)(r-t)$, hence $w > 1-w$ and thus $w > 1/2$. Under this constraint, it's easy to see that $F_w(a^q) > F_w(a^i)$ for all $i < q$. We indeed have $F_w(a^q) - F_w(a^i) = wq - (wi + (1-w)(q-i)) = (2w-1)(q-i) > 0$ since $w > 1/2$ and $q > i$. Hence a^q is necessarily an optimal solution in X. Note, however, that if the decision-maker was mistaken in the first answer ($a^t \succ a^r$ being the actual preference), then the same reasoning would have led to the choice of a^0. In this case, the recommendation a^q is in fact the worst possible recommendation given the actual DM's preferences.*

Although this example is a bit of a caricature, it does illustrate that a concern for efficiency in the active choice of a question to ask can lead to choices that are not robust to noisy responses. In this paper, we will propose a non-Bayesian approach to active learning of decision-maker preferences, which is more robust to noisy responses than usual methods based on regret minimization and enables us to identify or approximate a necessary winner in a given set of alternatives, as well as to build an explanatory model of decision-maker preferences. This is achieved by leveraging disagreement-based active learning, that we present in the following.

2 Noise-tolerant Active Preference Learning

Active learning [Tong and Koller, 2000, Balcan et al., 2006, Zhan et al., 2021, Cacciarelli and Kulahci, 2024] is a branch of machine learning that exploits the fact that, by carefully selecting which labeled inputs to use, it is sometimes possible to *achieve the same generalization performance* as in the *passive* setting (where models are trained on an i.i.d. dataset of labeled inputs) *while using fewer labeled examples.* In particular, we focus here on *disagreement-based active learning* [Dasgupta, 2011, Hanneke et al., 2014, Cortes et al., 2019, DeSalvo et al., 2021], which is a family of theoretically grounded active learning algorithms for binary classification. These algorithms share with incremental preference elicitation methods the common objective of reducing the space of admissible models as fast as possible in terms of the number of labeled inputs used (or number of asked queries to the DM using the preference elicitation terminology). In the following, we first present the basic disagreement-based active learning algorithms, and then we propose a disagreement-based active preference learning method for choice problems, which we illustrate on the toy case of Example 5.1.

2.1 Disagreement-based Active Learning

In binary classification, the data $\{(z^k, y^k)\}_{k=1}^t$ consists of inputs z^k belonging to an input space $\mathcal{Z}$ and their labels $y^k \in \mathcal{Y} = \{-1, +1\}$, assumed to be realizations of random variables $(Z^k, Y^k), k = 1, \dots, t$, i.i.d. according to a joint distribution over $\mathcal{Z} \times \mathcal{Y}$ denoted by $\mathcal{D}$. The marginal of $\mathcal{D}$ over $\mathcal{Z}$ is denoted by $\mathcal{D}_{\mathcal{Z}}$. Then, the goal of the learner is to identify within a hypothesis class $\mathcal{H}$ (a set of candidate classifiers), a classifier $h^* : \mathcal{Z} \to \mathcal{Y}$ that minimizes the expected probability of making an incorrect prediction, i.e.:

$$h^* \in \arg\min_{h \in \mathcal{H}} \mathbb{P}_{(Z,Y)\sim\mathcal{D}}(h(Z) \neq Y) \tag{5.4}$$

Predictor h^* is known as the *Bayes* predictor, and corresponds to the minimizer of the true risk (see Section 3.1.1 in Chapter 1) when prediction errors are computed with the 0-1 loss $l(\hat{y}, y) = \mathbb{1}[\hat{y} \neq y]$ since in this case $R(h) = \mathbb{E}_{(Z,Y)\sim\mathcal{D}}[\mathbb{1}[h(Z) \neq Y]] = \mathbb{P}_{(Z,Y)\sim\mathcal{D}}(h(Z) \neq Y)$.

In this setting, the general idea of disagreement-based active learning is embodied by the CAL algorithm [Cohn et al., 1994] (called after the authors' names Cohn, Atla and Ladner). Starting with $\mathcal{H}$ as the space of admissible models, CAL iteratively processes a sequence of unlabeled points $\{z^k\}_{k=1}^t$. At each iteration k, it *asks for the label, if and only if it does not have confidence in the answer* and if so, revise accordingly the space of models consistent with the new observed label. More precisely, let us denote by $\mathcal{H}_k$

the space of admissible models at iteration k such that $\mathcal{H}_0 = \mathcal{H}$. Then, at each iteration k, the learner asks for the true label of z^k if and only if there is a *disagreement* within $\mathcal{H}_k$ on its label, i.e., there exists two classifiers h_1,$h_2 \in \mathcal{H}_k$ such that $h_1(z^k) \neq h_2(z^k)$. The portion of the input space $\mathcal{Z}$ in which this condition holds defines the *disagreement region*. If asked, the newly obtained label y^k provides the additional constraint $h(z^k) = y^k$ on the set of admissible models, which now excludes the models classifying z^k differently, i.e., $\mathcal{H}_{k+1} = \{h \in \mathcal{H}_k | h(z^k) = y^k\}$. By doing so, the algorithm asks for a label if and only if the new constraint $h(z^k) = y^k$ surely reduces the space of admissible models, allowing to narrow down the version space around h^* with minimal labeling effort. Below, we illustrate the space of admissible models along with the disagreement region for a simple hypothesis class.

Example 5.2. *Let $\mathcal{Z} = \mathbb{R}^2$ and consider the hypothesis class $\mathcal{H} = \{h : \mathcal{Z} \to \mathcal{Y} | h_w(z) = \text{sign}(wz_1 + (1-w)z_2), w \in [0,1]\}$. Then, each positive (resp. negative) example induces the constraint $wz_1 + (1-w)z_2 \geq 0$ (resp. $wz_1 + (1-w)z_2 < 0$) on w, thus restricting $\mathcal{H}$. Let us now assume that the learner has received some positive and negative examples that are represented in Figure 5.3 (right) by blue and dashed red points respectively. They thus define an admissible space for w, i.e., to positively classify the point (-0.3 ,0.75) (blue filled point in Fig. 5.3 (right)) w must satisfy $0.3w + (1-w)0.75 \geq 0 \iff w \geq 0.71$, and to negatively classify the point (-0.75,0.25) (red filled point in Fig. 5.3 (right)) w must satisfy $-0.75w + (1-w)0.25 < 0 \iff w > 0.25$. This yields the interval of possible values represented in Figure 5.3 (left). Some linear separators of the form $wz_1 + (1-w)z_2$ for $w \in [0.25, 0.71]$ are represented by dotted lines in Figure 5.3 (right). Finally, the disagreement region is represented by the shaded area, in which, for any point (z_1, z_2), there always exists a weight w that would classify it positively and another that would classify it negatively. Thus, at the next iteration, the CAL algorithm will request the label of the new input if and only if it is located within this region.*

Obviously, CAL works to identify h^* under the hypothesis that it is compatible with any observed example, i.e., $h^*(z^k) = y^k$ (with probability 1) and thus $R(h^*) = 0$. In the more realistic noisy case where $R(h^*) > 0$ (often referred to as the *agnostic setting*), the hard constraints $h(z^k) = y^k$ will eventually exclude h^* from the set of admissible models. Extensions of the CAL algorithm in the noisy case [Balcan et al., 2006, Dasgupta et al., 2007] bypass this issue by defining the set of admissible models as the set of models that proved to yield small errors on the labeled examples. Among them, the DHM algorithm [Dasgupta et al., 2007] (also called after the authors' names Dasgupta, Hsu, and Monteleoni), which relies on supervised learning sub-tasks, provides a simple way of cautiously excluding models associated with significantly high empirical errors. In the

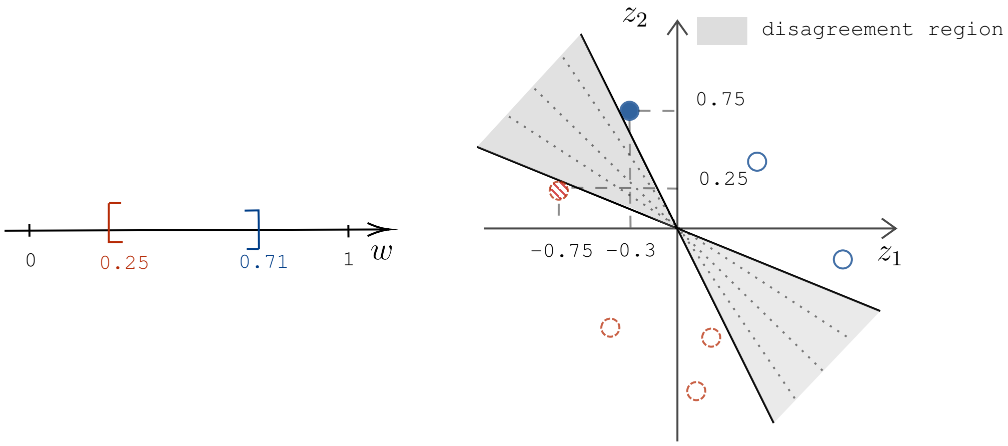


Figure 5.3: Set of admissible models (left) and disagreement region (right).

following, the empirical error of a model h on a set of labeled examples $S = \{(z^k, y^k)\}$ is denoted by $R_S(h)$ and defined as:

$$R_S(h) = \frac{1}{|S|} \sum_{(z'^\ell, y^k) \in S} \mathbb{1}[h(z^k) \neq y^k] \tag{5.5}$$

The DHM algorithm is given in Algorithm 5.2. It incrementally processes a sequence of unlabeled inputs $\{z^k\}_{k=1}^t$ and, at each iteration k, updates the set of labeled inputs, denoted by T_k, and the set of admissible models, denoted by $\mathcal{H}_k$. More precisely, at each iteration k, if there exists models in $\mathcal{H}_{k-1}$ assigning +1 to z^k (stored in $\mathcal{H}_k^+$) and others assigning -1 (stored in $\mathcal{H}_k^-$), the algorithm determines among them, the models h_k^+, h_k^- that yield the lowest empirical errors on the labeled inputs so far (i.e., T_{k-1}). Then, if $R_{T_{k-1}}(h_k^s) - R_{T_{k-1}}(h_k^{-s})$ exceeds a certain threshold Δ_k for $s \in \{+, -\}$, models in $\mathcal{H}_k^s$ yield empirical errors so high that it is unlikely h^* belongs to this set. In this case, the update $\mathcal{H}_k = \mathcal{H}_k^{-s}$ is thus applied and the label is not asked (*agreement* case). Remark that the notion of agreement is more flexible than in the CAL algorithm where it is required to have $\mathcal{H}_k^s = \emptyset$ for $s \in \{+, -\}$ to conclude to an agreement and not ask the label (this case is omitted in Algorithm 5.2 for the sake of clarity but it is naturally treated as in CAL i.e., $\mathcal{H}_k = \mathcal{H}_k^{-s} = \mathcal{H}_{k-1}$ and $T_k = T_{k-1}$). Otherwise, if h_k^+ and h_k^- have comparable errors on T_{k-1}, there is no clear evidence to determine whether h^* belongs to $\mathcal{H}_k^+$ or $\mathcal{H}_k^-$. Thus, $\mathcal{H}_k = \mathcal{H}_{k-1}$ and the label y^k of z^k is requested and stored in $T_k = T_{k-1} \cup \{(z^k, y^k)\}$ (*disagreement* case).

Let us now discuss in more details the calibration of the threshold Δ_k. Suppose that at some iteration k, we have $R_{T_{k-1}}(h_k^s) - R_{T_{k-1}}(h_k^{-s}) > \Delta_k$ for $s \in \{+, -\}$. Assume now that $h^* \in \mathcal{H}_k^s$. Thus, we have $R_{T_{k-1}}(h^*) \geq R_{T_{k-1}}(h_k^s)$ and $R_{T_{k-1}}(h^*) - R_{T_{k-1}}(h_k^{-s}) > \Delta_k$. Intuitively, if Δ_k represents an upper bound of the estimation error between the empirical risk difference $\delta_k = R_{T_{k-1}}(h^*) - R_{T_{k-1}}(h_k^{-s})$ and the true risk difference $\delta = R(h^*) - R(h_k^{-s})$

Algorithm 5.2: DHM

```
Inputs: $\mathcal{H}$, $\{z^k\}_{k=1}^t$
$\mathcal{H}_0 \leftarrow \mathcal{H}$ ;
$T_0 \leftarrow \{(z^0, y^0)\}$ ;
for $k = 1, \dots, t$ do
    $\mathcal{H}_k^+, \mathcal{H}_k^- \leftarrow \{h \in \mathcal{H}_{k-1} | h(z^k) = 1\}, \{h \in \mathcal{H}_{k-1} | h(z^k) = -1\}$ ;
    $h_k^+, h_k^- \leftarrow \arg\min_{h \in \mathcal{H}_k^+} R_{T_{k-1}}(h), \arg\min_{h \in \mathcal{H}_k^-} R_{T_{k-1}}(h)$;
    if $R_{T_{k-1}}(h_k^s) - R_{T_{k-1}}(h_k^{-s}) > \Delta_k$ for some $s \in \{+,-\}$ then
        $\mathcal{H}_k \leftarrow \mathcal{H}_k^{-s}$ , $T_k \leftarrow T_{k-1}$                #agreement;
    else
        $y^k \leftarrow$ label of $z^k$;
        $\mathcal{H}_k \leftarrow \mathcal{H}_{k-1}$, $T_k \leftarrow T_{k-1} \cup \{(z^k, y^k)\}$      #disagreement;
    $k \leftarrow k + 1$;
Outputs: $\mathcal{H}_k, T_k$
```

(i.e., such that $\delta_k \leq \delta + \Delta_k$ with high probability), we obtain $R(h^*) > R(h_k^{-s})$ with high probability. This is obviously contradictory the Bayes model definition (see Equation 5.4), and thus we can conclude that with high probability $h^* \notin \mathcal{H}_k^s$. Thus, Δ_k is calibrated to account for the estimation error between empirical and true risk differences. This estimation error naturally depends on the flexibility of the hypothesis class $\mathcal{H}$, that can be quantified using the *shatter coefficient* which measures the maximum number of ways $\mathcal{H}$ can label any set of k points, i.e.:

Definition 5.1 (shatter coefficient [Vapnik, 1995]). *For any hypothesis class $\mathcal{H}$ and $k \in \mathbb{N}$, the k^{th} shatter coefficient of $\mathcal{H}$ is denoted by $\mathcal{S}(\mathcal{H}, k)$ and defined by:*

$$\mathcal{S}(\mathcal{H}, k) = \max_{z_1, \dots, z_k \in \mathcal{Z}} |\{(h(z_1), \dots, h(z_k)) | h \in \mathcal{H}\}| \tag{5.6}$$

Then, threshold Δ_k in Algorithm 5.2 can be calibrated using the following result:

Lemma 5.1 (see Corollary 1 in [Dasgupta et al., 2007]). *For any $\delta > 0$ let $\beta_k = \sqrt{(4/k)\ln(8(k^2 + k)\mathcal{S}(\mathcal{H}, 2k)^2/\delta)}$, $h, h' \in \mathcal{H}_k$ and:*

$$\Delta_k(\beta_k, h, h') = \beta_k^2 + \beta_k \left(\sqrt{R_{T_{k-1}}(h)} + \sqrt{R_{T_{k-1}}(h')}\right) \tag{5.7}$$

Then, with probability at least $1 - \delta$, for any $k \geq 1$, :

$$R_{T_{k-1}}(h) - R_{T_{k-1}}(h') \leq R(h) - R(h') + \Delta_k(\beta_k, h, h') \tag{5.8}$$

Then, using Lemma 5.1, the following result sets the threshold value in Algorithm 5.2 so that the Bayes predictor is maintained in the set of admissible models. The proof is given to aid comprehension.

Lemma 5.2 (see Lemma 3 in [Dasgupta et al., 2007]). *For any $\delta > 0$, using $\Delta_k := \Delta_k(\beta_k, h_k^+, h_k^-)$ (see Equation 5.7) in Algorithm 5.2, we have with probability at least $1-\delta$ that for any $k \geq 1$:*

$$h^* \in \mathcal{H}_k$$

Proof. *We proceed by induction. At $k = 0$, $h^* \in \mathcal{H} = \mathcal{H}_0$ with probability 1. Assume now that $h^* \in \mathcal{H}_{k-1}$ with probability at least $1 - \delta$ for some $k \geq 1$. If $|R_{T_{k-1}}(h_k^+) - R_{T_{k-1}}(h^-)| \leq \Delta_k(\beta_k, h_k^+, h_k^-)$, then $\mathcal{H}_k = \mathcal{H}_{k-1} \ni h^*$ with probability at least $1 - \delta$. If $R_{T_{k-1}}(h_k^s) - R_{T_{k-1}}(h_k^{-s}) > \Delta_k(\beta_k, h_k^s, h_k^{-s})$ for $s \in \{+, -\}$, then $R_{T_{k-1}}(h_k^s) > \beta_k^2$. We assume now that $h^* \in \mathcal{H}_k^s$, then $R_{T_{k-1}}(h^*) \geq R_{T_{k-1}}(h_k^s)$, and thus we obtain:*

$$\begin{aligned}
R_{T_{k-1}}(h^*) - R_{T_{k-1}}(h_k^{-s}) &= R_{T_{k-1}}(h^*) - R_{T_{k-1}}(h_k^s) + R_{T_{k-1}}(h_k^s) - R_{T_{k-1}}(h_k^{-s}) \\
&\geq \sqrt{R_{T_{k-1}}(h_k^s)}\left(\sqrt{R_{T_{k-1}}(h^*)} - \sqrt{R_{T_{k-1}}(h_k^s)}\right) + R_{T_{k-1}}(h_k^s) - R_{T_{k-1}}(h_k^{-s}) \\
&> \beta_k\left(\sqrt{R_{T_{k-1}}(h^*)} - \sqrt{R_{T_{k-1}}(h_k^s)}\right) + \beta_k^2 \\
&+ \beta_k\left(\sqrt{R_{T_{k-1}}\left(h_k^s\right)} + \sqrt{R_{T_{k-1}}\left(h_k^{-s}\right)}\right) \\
&= \Delta_k(\beta_k, h^*, h_k^{-s})
\end{aligned}$$

Thus using Lemma 5.1, we obtain that with probability at least $1 - \delta$, $R(h^*) > R(h_k^{-s})$, which is contradictory with the definition of h^* (see Equation 5.4), and thus $h^* \notin \mathcal{H}_k^s$, i.e., $h^* \in \mathcal{H}_k^{-s} = \mathcal{H}_k$ since $h^* \in \mathcal{H}_{k-1}$ by induction hypothesis.

In the next section, we propose to exploit and extend DHM to the multicriteria choice problem under noisy answers.

2.2 A Disagreement-based Active Preference Learning Algorithm

We now introduce an algorithm to solve the choice problem with noisy answers using disagreement-based active learning. It is designed to achieve a twofold objective: on the one hand, quickly finding a near-optimal solution within X, and on the other hand, assessing parameter w to have a predictive model F_w of DM's preferences. To this end, the proposed algorithm combines DHM (see Algo. 5.2) with a regret control mechanism as in IPE (see Algo. 5.1). Before giving the algorithm, we first explicit how preference learning from pairwise comparisons fits into the framework of binary classification.

Preference learning as binary classification In the multicriteria choice problem setting, the determination of the weight vector $w \in \mathcal{W}$ in the preference model F_w from pairwise preference examples $x^\ell \succsim x'^\ell$, $x^\ell, x'^\ell \in X$ can be formulated as a binary classification problem where $z^\ell = (x^\ell, x'^\ell) \in \mathcal{Z} = X^2$ and $y^\ell = 1$ if $x^\ell \succsim x'^\ell$ and $y^\ell = -1$ otherwise ($x^\ell \prec x'^\ell$). In this case, the hypothesis class can be defined as follow:

$$\mathcal{H} = \{h_w : X^2 \to \mathbb{R} | h_w(x, x') = \text{sign}(F_w(x) - F_w(x')), w \in \mathcal{W}\} \tag{5.9}$$

Thus, $\mathcal{H}$ can be identified with the set of possible preferences induced by $w \in \mathcal{W}$ trough model F_w. In this set, w^* denotes the weight vector that best represents the DM's preferences, i.e., such that h_{w^*} is the Bayes predictor (see Eq. 5.4). Thus w^* is referred to as the Bayes weight in what follows. Furthermore, to ease notation, the empirical error of h_w on a labeled set S of examples (i.e., $R_S(h_w)$; see Eq. 5.5), is now denoted by $R_S(w)$.

The proposed algorithm We propose to simulate a stream of examples of unlabeled pairs of alternatives (x^k, x'^k) independently and uniformly drawn from X, and to apply the DHM algorithm to assess whether the labels are worth querying or not, and progressively reduce the set of admissible weights without excluding the Bayes weight until a solution with a sufficiently low maximum regret (see equation 5.2) emerges and can be recommended. Using $\mathcal{H}$ of the form of Equation 5.9 and $\mathcal{H}_k := \mathcal{W}_k$ (and $\mathcal{H}_k^+, \mathcal{H}_k^- := \mathcal{W}_k^+, \mathcal{W}_k^-$) at any iteration k in the DHM algorithm (see Algorithm 5.2), we propose Algorithm 5.3.

To aid understanding of Algorithm 5.3, we now describe and illustrate it. It takes as input the set of alternatives X, an initial set of admissible weight vectors $\mathcal{W}_0$, and a desired reduction percentage ρ of the maximum regret of the recommended solution (compared to the first iteration). Then, it sequentially proceeds pairs of alternatives x^k, x'^k drawn randomly and uniformly from the set of alternatives X, and at iteration k, asks for the DM to provide an answer y^k to the pairwise comparison query *$x^k \succsim x'^k$?* if and only if models within $\mathcal{W}_{k-1}$ *disagree* on the label.

More precisely, if $w_k^+ = \arg\min_{w \in \mathcal{W}_k^+} R_{T_{k-1}}(w)$, $w_k^- = \arg\min_{w \in \mathcal{W}_k^-} R_{T_{k-1}}(w)$ and $|R_{T_{k-1}}(w_k^+) - R_{T_{k-1}}(w_k^-)|$ is lower than a certain threshold Δ_k, the elements of $\mathcal{W}_{k-1}$ somehow *disagree* on whether $x^k \succ x'^k$ or $x^k \precsim x'^k$. Indeed, the weight vectors verifying $x^k \succsim x'^k$ ($\mathcal{W}_k^+$) and the weight vectors verifying $x^k \prec x'^k$ ($\mathcal{W}_k^-$) are attached to similar minimal errors on the learning database T_{k-1}. Therefore, the answer y^k is likely to provide new information, and thus the query $x^k \succsim x'^k$? is asked to the DM. Then, the answer y^k is stored as a new preference example (x^k, x'^k, y^k) in the learning database, i.e., $T_k = T_{k-1} \cup \{(x^k, x'^k, y^k)\}$. This case is illustrated in Figure 5.4.

However, if $|R_{T_{k-1}}(w_k^+) - R_{T_{k-1}}(w_k^-)| > \Delta_k$, the elements of $\mathcal{W}_{k-1}$ somehow *agree* on

Algorithm 5.3: Active preference learning

```
Inputs: $\mathcal{W}_0, X, \rho$
draw a pair $(x^0, x'^0)$ uniformly in $X^2$
$y^0 \leftarrow$ answer to the query $x^0 \succsim x'^0$ ?
$T_0 \leftarrow \{(x^0, x'^0, y^0)\}, \mathrm{MR}_0 \leftarrow 1$, $k \leftarrow 1$
while $\mathrm{MR}_{k-1} > \rho \mathrm{MR}_0$ do
    draw a pair $(x^k, x'^k)$ uniformly in $X^2$
    $\mathcal{W}_k^+, \mathcal{W}_k^- \leftarrow \{w \in \mathcal{W}_{k-1} | F_w(x^k) \geq F_w(x'^k)\}$,$\{w \in \mathcal{W}_{k-1} | F_w(x^k) < F_w(x'^k)\}$
     if $\mathcal{W}_k^+ \neq \emptyset$ and $\mathcal{W}_k^- \neq \emptyset$ then
        $w_k^+, w_k^- \leftarrow \arg\min_{w \in \mathcal{W}_k^+} R_{T_{k-1}}(w), \arg\min_{w \in \mathcal{W}_k^-} R_{T_{k-1}}(w)$
    if $\mathcal{W}_k^- \neq \emptyset$ and $(\mathcal{W}_k^+ = \emptyset$ or $R_{T_{k-1}}(w_k^+) - R_{T_{k-1}}(w_k^-) > \Delta_k)$ then
        $\mathcal{W}_k \leftarrow \mathcal{W}_k^-$ , $T_k \leftarrow T_{k-1}$
    else if $\mathcal{W}_k^+ \neq \emptyset$ and $(\mathcal{W}_k^- = \emptyset$ or $R_{T_{k-1}}(w_k^-) - R_{T_{k-1}}(w_k^+) > \Delta_k)$ then
        $\mathcal{W}_k \leftarrow \mathcal{W}_k^+$ , $T_k \leftarrow T_{k-1}$
    else
        $y^k \leftarrow$ answer to the query $x^k \succsim x'^k$ ?;
        $\mathcal{W}_k \leftarrow \mathcal{W}_{k-1}$,$T_k \leftarrow T_{k-1} \cup \{(x^k, x'^k, y^k)\}$
    $\hat{w}_k \leftarrow \arg\min_{w \in \mathcal{W}_k} R_{T_k}(w)$
    $\hat{x}_k, \mathrm{MR}_k \leftarrow \arg\max_{x \in X} F_{\hat{w}_k}(x), \mathrm{MR}(\hat{x}_k, \mathcal{W}_k)$
    $k \leftarrow k + 1$
Outputs: $\hat{w}_{k-1}$,$\hat{x}_{k-1}$, $\mathrm{MR}_{k-1}$
```

whether $x^k \succsim x'^k$ or $x^k \precsim x'^k$, since one of the two sets $\mathcal{W}_k^+, \mathcal{W}_k^-$ yields significantly higher errors on the learning database T_{k-1} than the other, and thus is not likely to contain the best predictor w^*. Then, the query is not worth asking and the algorithm exploits the agreement of $\mathcal{W}_{k-1}$ on the instance (x^k, x'^k) to reduce, with high confidence, the set of admissible models by adding the preference as a hard constraint, i.e., $\mathcal{W}_k = \mathcal{W}_k^+$, if $\mathcal{W}_k^+$ yields the smallest error, and $\mathcal{W}_k = \mathcal{W}_k^-$ otherwise. In this case, no example is added to the learning database, i.e., $T_k = T_{k-1}$. It is illustrated in Figure 5.5.

This cautious reduction of the set of admissible weights makes it possible to incrementally control the remaining level of uncertainty on the DM's best alternative in X, as in incremental preference elicitation (see Algorithm 5.1). Indeed, at the end of iteration k, the learned model is $\hat{w}_k = \arg\min_{w \in \mathcal{W}_k} R_{T_k}(w)$ and naturally the recommended solution is $\hat{x}_k = \arg\max_{x \in X} F_{\hat{w}_k}(x)$. Then, the remaining level of uncertainty on the DM's best alternative is assessed by computing the maximum regret attached to the recommendation of $\hat{x}_k$ knowing that the current set of admissible weights is $\mathcal{W}_k$, i.e., $\mathrm{MR}_k = \mathrm{MR}(\hat{x}_k, \mathcal{W}_k)$. Once MR_k is sufficiently reduced (ratio $\mathrm{MR}_k \,/\, \mathrm{MR}_1$ below threshold $\rho \in [0, 1)$), the algorithm stops and outputs a recommended solution $\hat{x}_k$.

It is important to note that at each iteration, Algorithm 5.3 requires minimizing $R_{T_k}(w)$ over $\mathcal{W}_k^+$, $\mathcal{W}_k^-$ and $\mathcal{W}_k$ and (see line 8 and 17 in Algorithm 5.3). However, function $w \mapsto R_S(w)$ is non-convex and discontinuous for any set of labeled examples S, making

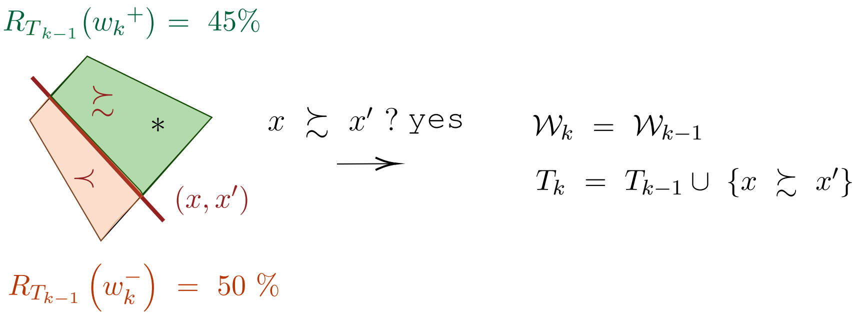

$R_{T_{k-1}}(w_k{}^+) = 45\%$
$\succsim$
$*$
$\prec$
(x, x')
$x \succsim x'$? yes
$\mathcal{W}_k = \mathcal{W}_{k-1}$
$T_k = T_{k-1} \cup \{x \succsim x'\}$
$R_{T_{k-1}}\left(w_k^-\right) = 50\ \%$

Figure 5.4: Illustration of the *disagreement* case.

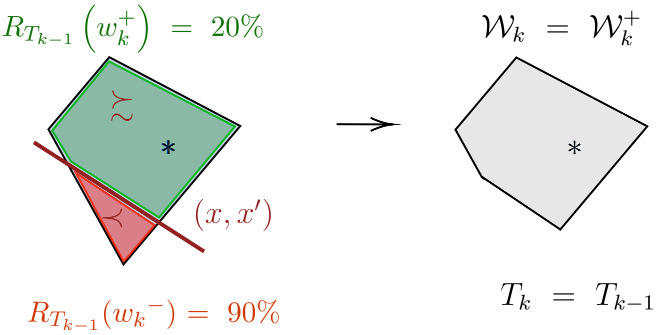

$R_{T_{k-1}}\left(w_k^+\right) = 20\%$
$\succsim$
$*$
$\prec$
(x, x')
$R_{T_{k-1}}(w_k{}^-) = 90\%$
$\mathcal{W}_k = \mathcal{W}_k^+$
$*$
$T_k = T_{k-1}$

Figure 5.5: Illustration of the *agreement* case.

its optimization intractable. Here, to bypass this issue, we start with a set of admissible models $\mathcal{W}_0$ that is a generated finite approximation of $\mathcal{W}$ such that $\mathcal{W}_0 \subseteq \mathcal{W}$ (obtained for instance with a uniform sampling of $\mathcal{W}$), and solve the optimization tasks by exhaustive search on $\mathcal{W}_0$ (or $\mathcal{W}_k$ later on in the algorithm). An important consequence follows from this choice: as no optimization task is performed over parameter w the algorithm applies to a broad class of aggregation functions F_w, including not only linear but also non-linear functions in their parameters, such as the Chebyshev norm (see Definition 1.16), for instance.

Then, using Lemma 5.2, we establish Proposition 5.1 showing how threshold values Δ_k can be set to make sure that $\mathcal{W}_k, k \geq 1$ contain with high probability the Bayes weight vector w^*. It also shows that with high probability MR_k upper bounds the regret of the recommended solution under w^*, i.e., $\max_{x \in X}\{F_{w^*}(x) - F_{w^*}(\hat{x}_k)\}$ and thus is a sound indication of the remaining level of uncertainty on the recommended solution.

Proposition 5.1. *For $\delta > 0$, $\gamma_k = \sqrt{(4/k)\ln\left(8\left(k^2+k\right)|\mathcal{W}_0|^2/\delta\right)}$ and $\Delta_k := \Delta_k(\gamma_k, w_k^+,$ $w_k^-)$ given by Equation 5.7, with probability at least $1-\delta$, we have for any $k \geq 1$ in Algorithm 5.3:*

$$w^* \in \mathcal{W}_k \tag{5.10}$$

and MR_k *upper bounds the real regret, i.e.:*

$$\mathrm{MR}_k \geq \max_{x \in X}\{F_{w^*}(x) - F_{w^*}(\hat{x}_k)\} \tag{5.11}$$

where $\hat{x}_k = \arg\max_{x\in X} F_{\hat{w}_k}(x)$ *is the recommended solution at iteration* k*.*

Proof. Algorithm 5.3 is Algorithm 5.2 (DHM) with a modified stopping criterion for $\mathcal{H} = \{h_w : X^2 \to \mathbb{R} | h_w(x,x') = \mathrm{sign}(F_w(x) - F_w(x')), w \in \mathcal{W}_0\}$ *and a dataset of samples* (x^k, x'^k) *i.i.d. according to the uniform distribution over* X^2*. Then, by Lemma 5.2, for* $\Delta_k := \Delta_k(\beta_k, w_k^+, w_k^-)$ *given by Equation 5.7 with* $\beta_k = \sqrt{(4/k)\ln(8\,(k^2+k)\,\mathcal{S}(\mathcal{H},2k)^2}$*, with probability at least* $1-\delta$*, we have* $w^* \in \mathcal{W}_k$*, for any* $k \geq 1$*. Thus, it is sufficient to show that Lemma 5.2 is still verified using* $\gamma_k = \sqrt{(4/k)\ln(8\,(k^2+k)\,|\mathcal{W}_0|^2/\delta)}$*.*

Firstly, by definition of the shatter coefficient (see Definition 5.1), for any $k \geq 1$*:*

$$\mathcal{S}(\mathcal{H}, 2k) \leq |\mathcal{H}| \leq |\mathcal{W}_0| \tag{5.12}$$

Thus, $\gamma_k \geq \beta_k$ *and by Equation 5.7,* $\Delta_k(\gamma_k, w_k^+, w_k^-) \geq \Delta_k(\beta_k, w_k^+, w_k^-)$*. Let us now denote* $\mathcal{W}_k$ *and* $\mathcal{W}'_k$ *the sets of admissible hypothesis at iteration* k *in Algorithm 5.3 with* $\Delta_k := \Delta_k(\gamma_k,, w_k^+, w_k^-)$ *and* $\Delta_k := \Delta_k(\beta_k, w_k^+, w_k^-)$ *respectively. Then we show that* $\mathcal{W}'_k \subseteq \mathcal{W}_k$ *for any* $k \geq 0$ *(with probability 1). We proceed by induction. At* $k = 0$*,* $\mathcal{W}'_0 = \mathcal{W}_0$*. Assuming that* $\mathcal{W}'_{k-1} \subseteq \mathcal{W}_{k-1}$ *for some* $k \geq 1$*, at the next iteration, if* $|R_{T_{k-1}}(w_k^+) - R_{T_{k-1}}(w_k^-)| \leq \Delta_k(\beta_k) \leq \Delta_k(\gamma_k)$*, then* $\mathcal{W}'_k = \mathcal{W}'_{k-1} \subseteq \mathcal{W}_{k-1} = \mathcal{W}_k$*. If* $\Delta_k(\beta_k) < |R_{T_{k-1}}(w_k^+) - R_{T_{k-1}}(w_k^-)| \leq \Delta_k(\gamma_k)$*, then* $\mathcal{W}'_k = \mathcal{W}'_{k-1} \setminus \mathcal{W}'^s_k$ *for some* $s \in \{+,-\}$ *and* $\mathcal{W}_{k-1} = \mathcal{W}_k$*, and thus* $\mathcal{W}'_k \subseteq \mathcal{W}'_{k-1} \subseteq \mathcal{W}_{k-1} = \mathcal{W}_k$*. Finally if* $\Delta_k(\beta_k) \leq \Delta_k(\gamma_k) < |R_{T_{k-1}}(w_k^+) - R_{T_{k-1}}(w_k^-)|$*, then* $\mathcal{W}'_k = \mathcal{W}'_{k-1} \setminus \mathcal{W}'^s_k \subseteq \mathcal{W}_{k-1} \setminus \mathcal{W}^s_k = \mathcal{W}_k$ *for some* $s \in \{+,-\}$*. Therefore, in any case,* $\mathcal{W}'_k \subseteq \mathcal{W}_k$*. Intuitively, the higher the threshold, the more cautious the algorithm and the slower the reduction of the set of admissible sets. Then,* $w^* \in \mathcal{W}'_k \subseteq \mathcal{W}_k$ *for any* $k \geq 1$ *with probability at least* $1-\delta$*. Thus, with probability at least* $1-\delta$*,* $\max_{x\in X}\{F_{w^*}(x) - F_{w^*}(\hat{x}_k)\} \leq \max_{w\in\mathcal{W}_k}\max_{x\in X}\{F_w(x) - F_w(\hat{x}_k)\} = \mathrm{MR}_k$*.*

In the next section dedicated to numerical tests, we will see that $\Delta_k(\gamma_k, w_k^+, w_k^-)$ is an over-cautious threshold, that may induce many preference queries. In practice, a more aggressive threshold $\alpha\Delta_k(\gamma_k, w_k^+, w_k^-)$ where $\alpha \in [0,1]$ can be used to reduce the number of preference queries without too much sacrificing the recommendation quality. Parameter α can be set using cross-validation on a small test set of preference examples $\{(x^k, x'^k, y^k)\}$. On the other side, parameter δ is set to 0.95.

2.3 Illustration on a Toy Example

To illustrate the benefit of Algorithm 5.3 for preference elicitation with noisy answers, we exploit the easy-to-grasp toy case of Example 5.1 on the choice set $X = \{a^0, \dots, a^q\}$ illustrated in Figure 5.2. The first pair under consideration is $(x^0, x'^0) = (a^r, a^t)$ with $r > t$. Since the first DM's preference statement is $a^r \succ a^t$, Algorithm 5.3 starts with the initial learning database $T_0 = \{(x^0, x'^0, 1)\}$. Then, examples of pairs $(x^k, x'^k) = (a^{r_k}, a^{t_k})$ are repeatedly drawn uniformly from X^2 (where we always have $r_k \geq t_k$ without loss of generality). Algorithm 5.3 then either asks the query $a^{r_k} \succsim a^{t_k}$*?* or if confident enough to predict the DM's answer, does not ask the query and updates the current set of admissible weights accordingly to either $\mathcal{W}_k^+ = [\frac{1}{2}, 1]$ or $\mathcal{W}_k^- = [0, \frac{1}{2})$ (if $r_k > t_k$). In this case, the recommended alternative $\hat{x}_k = \arg\max_{x \in X} F_{\hat{w}_k}(x)$ with $\hat{w}_k = \arg\min_{w \in \mathcal{W}_k} R_{T_k}(w)$ is necessarily a^q if $\mathcal{W}_k = [\frac{1}{2}; 1]$ or a^0 if $\mathcal{W}_k = [0; \frac{1}{2})$. Then, in any case, $\mathrm{MR}(\hat{x}_k) = 0$ and the algorithm stops. Indeed, for instance if $\mathcal{W}_k = [\frac{1}{2}; 1]$, we have:

$$\max_{r \in \{0,\dots,q\}} \max_{w \in [1/2,1]} (F_w(a^r) - F_w(a^q)) = \max_{r \in \{0,\dots,q\}} \max_{w \in [1/2,1]} \{(r-q)(2w-1)\} = 0$$

Also, at each iteration k such that $r_k > t_k$, it can easily be checked that:

$$\begin{aligned} R_{T_{k-1}}(w_k^+) &= \min_{w \in [1/2,1]} R_{T_{k-1}}(w) \\ &= \min_{w \in [1/2,1]} \frac{1}{|T_{k-1}|} \sum_{i=0}^{k-1} \mathbb{1}[F_w(a^{r_i}) - F_w(a^{t_i}) \neq y^i] \\ &= \frac{1}{|T_{k-1}|} \sum_{i=0}^{k-1} \mathbb{1}[1 \neq y^i] \end{aligned}$$

A similar reasoning yields $R_{T_{k-1}}(w_k^-) = \frac{1}{|T_{k-1}|} \sum_{i=0}^{k-1} \mathbb{1}[-1 \neq y^i]$. Thus $R_{T_{k-1}}(w_k^-)$ and $R_{T_{k-1}}(w_k^+)$ are respectively the frequencies of occurrences of preferences of type $a^{r_i} \succsim a^{t_i}$ and of type $a^{r_i} \prec a^{t_i}$ in the sequence of the past DM's preference statements. Thus, Algorithm 5.3 recommends a solution when one of the two frequencies becomes significantly higher than the other, i.e., with a difference higher than $\alpha\Delta_k(\gamma_k, w_k^+, w_k^-)$ which, according to Proposition 5.1, is decreasing with k. This choice process is more robust than taking for granted the very first DM's preference statement, as illustrated below.

In Table 5.1, we present numerical results obtained for the described toy case with $q = 20$. To assess the benefit of Algorithm 5.3 in the context of noisy answers we introduce a random noise in the simulated DM's answers which swaps the answers with probability $p = 0.1$ and then $p = 0.3$. Parameters α and ρ of Algorithm 5.3 are respectively set to

	$p = 0.1$		$p = 0.3$	
	Nb. of queries	Rec. accuracy	Nb. of queries	Rec. accuracy
Algorithm 5.1	1.0	89%	1.0	74%
Algorithm 5.3	3.5	99%	14.0	90%

Table 5.1: Comparison of Algorithm 5.3 and Algorithm 5.1 over 100 simulations.

$\alpha = 0.05$ and $\rho = 0$. For the two noise levels, we compare Algorithm 5.3 to Algorithm 5.1 on 500 simulations in terms of number of queries and accuracy of the recommendation (number of simulations where the recommendation was correct). Looking at the results, we can see that Algorithm 5.1, while always terminating after one query, suffers from noisy answers and does recommend the optimal alternative in only 89% of the time for the low noise level, and 74% of the time for the high noise level. On the contrary, Algorithm 5.3 recommends the optimal alternative in nearly 100% of the time for the low noise level while asking only 3.5 questions on average, and in 90% of the time for the high noise level with about 14 questions on average. Further tests are conducted in the next section.

3 Numerical Tests

In this section, we present the results of numerical tests performed on synthetic preference data. We test the ability of Algorithm 5.3 to provide accurate recommendations while receiving noisy answers from the DM, and when possible, we compare those results to Algorithm 5.1. The tests are conducted with F_w taken as the weighted sum, the 2-additive Choquet integral and the Chebyshev distance. We consider random finite sets of admissible models $\mathcal{W}_0 \subseteq \mathcal{W}$ of size $|\mathcal{W}_0| = 5000$ for each experiment. For the weighted sum and the Chebyshev distance to the ideal point, $\mathcal{W}_0$ is obtained by uniform sampling of the simplex $\mathcal{W} = \{w \in [0,1]^n | \sum_{i=1}^n w_i = 1\}$. For the 2-additive Choquet integrals, the set of admissible weights is the set of capacities $\mathcal{W} = \{w : 2^N \to \mathbb{R} | A \subseteq N, w(A) \leq w(B), w(\emptyset) = 0, w(N) = 1\}$, restricted to 2-additive capacities. The set of 2-additive capacities is a polyhedron admitting a polynomial number of extreme points [Grabisch et al., 2016] (Theorem 2.65), namely, the *unanimity games* defined for any $i \in [\![1, \frac{n(n+1)}{2}]\!]$ by:

$$v_i(S) = \begin{cases} 1 & \text{if } Y_i \subseteq S \\ 0 & \text{otherwise} \end{cases}$$

where $Y_i \subseteq N$ is any nonempty subset of size at most 2, and the *conjugates of unanimity* *games* defined for any $i \in [\![\frac{n(n+1)}{2} + 1, n^2]\!]$ by:

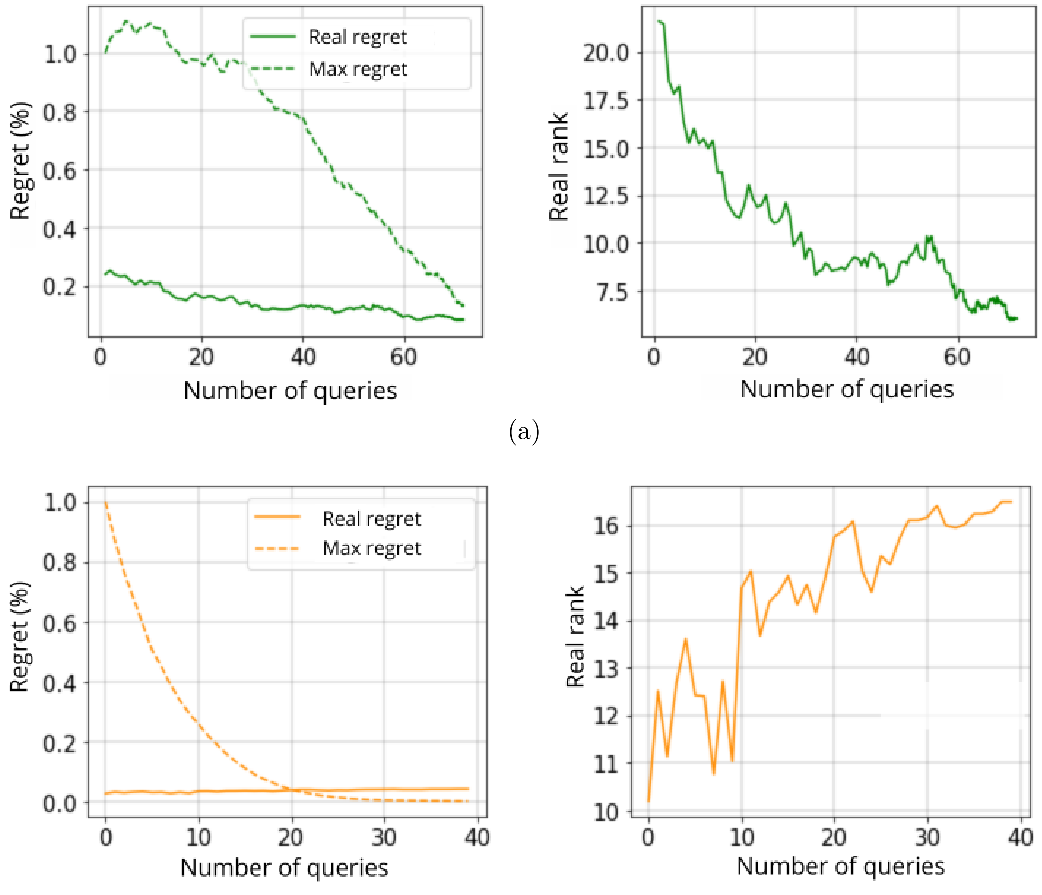
$$v_i(S) = \begin{cases} 1 & \text{if } Y_i \cap S \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

Hence, any 2-additive capacity w can be generated by a convex combination $w = \sum_{i=1}^{q} \beta_i v_i$ with $q = n^2$,$\beta_i \in [0, 1]$, $\sum_{i=1}^{q} \beta_i = 1$. Thus, we obtain samples $\mathcal{W}_0$ of the set of 2-additive capacities by uniform sampling of the simplex $\{\beta \in [0, 1]^q | \sum_{i=1}^{q} \beta_i = 1\}$.

For all the experiments, the DM's answers are simulated according to a ground truth model $F_{w_{gt}}$ with a random weight vector w_{gt} generated in the same way as the elements of $\mathcal{W}_0$. The answers are disturbed with random noises ϵ such that $y^k = \text{sign}(F_{w_{gt}}(x) - F_{w_{gt}}(x') + \epsilon_k)$ and ϵ_k is uniformly distributed within $[-\sigma, \sigma]$ with noise level $\sigma > 0$, for any k.

In the first experiment, we compare the noise tolerance of Algorithm 5.3 and Algorithm 5.1 when F_w is the Choquet integral, $n = 5$, and $|X| = 100$ (containing solely Pareto-optimal solutions). Parameter α of Algorithm 5.3 is set to $\alpha = 1.7 \times 10^{-2}$. Figures 6.7 and 6.8 respectively show, for Algorithm 5.3 and Algorithm 5.1, the average rank in the DM's hidden ranking (right) and regret (left) of the recommended solution over 100 simulations w.r.t. the number of query. More precisely, 6.7(left) shows the *real* regret of the recommended solution by Algorithm 5.3, i.e., $\max_{x' \in X}\{F_{w_{gt}}(x') - F_{w_{gt}}(\hat{x}_k)\}$ (plain green line) along with the upper bound MR_k (dotted green line). Note that both values are represented in percentage w.r.t. MR_1. Then, in Figure 6.8(left) is represented the *real* regret of the recommended solution by Algorithm 5.1 (plain orange line) along with the min-max regret (i.e., $\text{mMR}(X, \mathcal{W}_k)$). Note that both values are represented in percentage w.r.t. $\text{mMR}(X, \mathcal{W}_0)$. In Figure 6.8, we observe that while the min-max regret quickly reduces with Algorithm 5.1, it does not induce a reduction of the real regret and yields recommended solutions with increasing real ranks. On the contrary, in Figure 6.7, we observe that while decreasing more slowly, the bound MR_k of Algorithm 5.3 decreases accordingly with the real regret and rank of the recommended solution $\hat{x}_k$. After 30 queries, the real rank of the recommended solution is about 8 for Algorithm 5.3 and 16 for Algorithm 5.1.

In the second experiment, we show different tradeoffs between quality of the recommendation and number of asked queries that can be achieved with Algorithm 5.3 by varying the α parameter, which controls the threshold value $\alpha\Delta_k$. The tests are conducted for the Chebyshev distance for $n = 10$, $X = 100$ (containing solely Pareto-optimal solutions) and α varying in a uniform grid within $[5 \times 10^{-3}, 5 \times 10^{-2}]$. The results are averaged over 100 simulations, and for this experiment, the DM's answers are disturbed



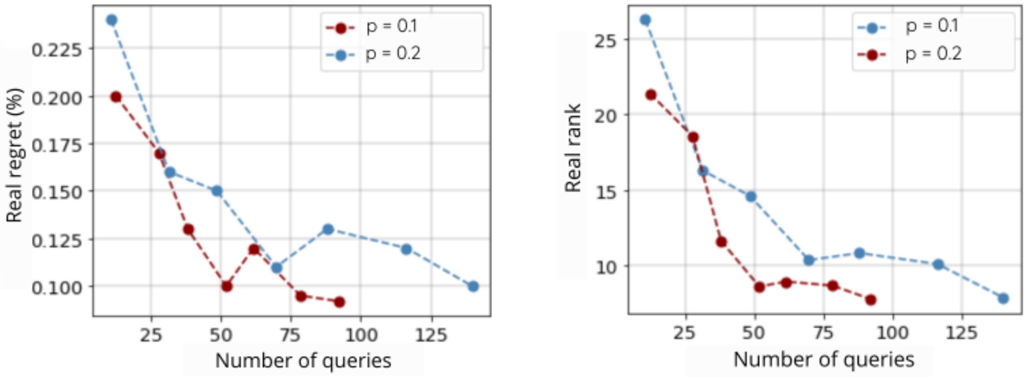
(a)

(b)

Figure 5.6: Real regret and real rank for Algorithm 5.3 (a) and Algorithm 5.1 (b).

with a random noise which swaps the answers with probability $p = 0.1$ and then $p = 0.2$. For both noise levels, Figure 5.7a (left) represents the average real regret of the recommended solutions (in percentage w.r.t. MR_1) versus the average number of asked queries and Figure 5.7a (right) shows the average real rank of the recommended solution, again versus the average number of queries. For all figures, the higher the α value, the higher the caution level of Algorithm 5.3, and thus the higher the number of asked queries. For $p = 0.1$ (red), asking 7 queries yields a real regret of 20% in average with an average real rank equal to 21 and asking 50 queries reduces the real regret to 10% and the average rank to 8. When the noise level increases, the performances weaken. For instance, for $p = 0.2$, 7 questions yield an average real rank equal to 26.

In the third experiment, we compare Algorithm 5.3 to another non-Bayesian active learning method recently proposed for linear models [Pourkhajouei et al., 2023, Escamocher et al., 2025]. This method also exploits the idea of minimizing the 0-1 loss error on the set of admissible models $\mathcal{W}$ instead of irreversibly reducing the set of admissible models such as in Algorithm 5.1. However, while being effective at solving choice prob-



(a) (b)

Figure 5.7: Real regret (a) and rank (b) w.r.t. query number with Algorithm 5.3.

lems with small number of queries, the used querying strategy focuses only on the most plausible best elements of X, and thus, the learned model shows lower generalization performances on $\mathcal{X}$ and is farther from the hidden model w_{gt} than the one obtained with Algorithm 5.1. This can be seen in Table 5.2 where both methods are compared in terms of query number and real rank of the recommended solutions; we also give the average absolute distance to w_{gt} of the learned preference model and the test accuracy defined as the percentage of preference inversion on a test set of pairwise comparison in $\mathcal{X}$. The tests are conducted with F_w taken as the weighted sum, $n = 10$, $\sigma = 0.05$, $|X| = 1000$ (containing solely Pareto-optimal solutions) and for Algorithm 5.3 the parameter are set to $\alpha, \rho = (2 \times 10^{-10}, 0.5)$ which allow yielding similar query numbers for both methods. We observe that while yielding similar results in terms of query number and real rank, Algorithm 5.3 better recovers preference model w_{gt} and achieves a higher test accuracy. Computation times are comparable for both methods (3.35 sec. for Algorithm 5.3 and 1.33 sec for [Pourkhajouei et al., 2023] on average).

	Query number	Real rank	Distance to w_{gt}	Test accuracy
Algorithm 5.3	68.0 ± 22.7	$40.1/1000 \pm 59.8$	0.03 ± 0.01	$88.5\% \pm 2.5\%$
[Pourkhajouei]	60.9 ± 17.3	$44.75/1000 \pm 81.13$	0.07 ± 0.02	$78.0\% \pm 5.0\%$

Table 5.2: Comparison with [Pourkhajouei et al., 2023] over 100 simulations.

4 Conclusion

We have presented a new approach for determining an optimal solution in a given set, by actively learning the parameters of an aggregation function describing the DM's preferences. This approach is a cautious version of the standard Algorithm 5.1 based on the minimax regret criterion that progressively reduces the set of admissible model parameters, until a zero-regret (or near-zero-regret) solution appears as a necessary winner. In our view, our approach offers three significant advantages.

Firstly, it is more error-tolerant, since the DM's responses are not systematically interpreted as hard constraints on the parameter space. The numerical tests carried out in Section 4 clearly demonstrated the gain in robustness in the face of noisy responses. The second advantage is that, beyond the identification of an optimal choice, the method provides a learned model that can be used to explain decisions and make choices on new instances. Finally, it does not require the scalarizing function to be linear in its parameters and thus applies to a wider class of aggregators, including the weighted Chebyshev norm, or the Sugeno integral that is generally not learned by regret minimization.

Algorithm 5.3 also brings some advantages compared to recently proposed approaches for preference learning with noisy DM's answers, whether Bayesian [Chajewska et al., 2000, Bourdache et al., 2019a] or non-Bayesian [Pourkhajouei et al., 2023]. On the one hand, being non-Bayesian, the proposed approach does not require knowledge of a prior distribution on the model parameters, a strong assumption often necessary to initiate Bayesian learning. On the other hand, concerning non-Bayesian approaches, the numerical tests presented at the end of Section 4 show that Algorithm 5.3, while exhibiting comparable performance to recent alternative proposals [Pourkhajouei et al., 2023] in terms of robustness to noisy responses, achieves significantly better generalization performance and thus is likely to make better decision on new instances of choice problems.

An interesting research direction would be to no longer start from a discrete subset $\mathcal{W}_0$ of the set of admissible weights $\mathcal{W}$, but to work directly with the latter, in order to fully exploit the richness of the chosen aggregation function. This requires addressing two challenges:

- The hypothesis class $\mathcal{H}$ associated with $\mathcal{W}$ (see Equation 5.9) is no longer discrete. Thus, its shatter coefficient is known to be in $O(k^d)$ (Sauer's Lemma [Sauer, 1972]) where d is its *Vapnik-Chervonenkis (VC) dimension*, which corresponds to the maximum number of points k such that $\mathcal{S}(k, \mathcal{H}) = 2^k$ (i.e., every possible labeling of these points can be realized by some hypothesis in $\mathcal{H}$). Therefore, deriving the VC dimension of the hypothesis class associated with the different aggregation functions is necessary to compute the threshold Δ_k used in Algorithm 5.3. More specifically, as shown in Proposition 5.1, an upper bound is sufficient. Some contributions have addressed this question for the Choquet integral [Hüllermeier and Fallah Tehrani, 2012, Basu and Echenique, 2020]. In particular, by leveraging the fact that any linear classifier of dimension d has a VC dimension of $d+1$ (see for instance [Vapnik, 1995]; Chapter 3) and that, using the Möbius transform, the Choquet integral expresses as a linear model in a space of dimension $2^n - 1$ (see Equation 1.5), an upper bound in $O(2^n)$ can be obtained. However, whether a tighter bound exists remains an open question (this is critical for Algorithm 5.3, as a looser upper bound results in a higher number of queries to the DM). Also, to the best of our knowledge, no contributions address this question for non-linear aggregation functions such as the Chebyshev norm or the Sugeno integral.

- Avoiding the discretization of the space $\mathcal{W}$ means having to optimize the 0-1 loss over $\mathcal{W}$ in Algorithm 5.3. One way to avoid this typically intractable optimization task is to replace it with a convex surrogate loss, such as the hinge loss $l(y, h(x)) = \max\{0, 1-h(x)y\}$. Then, as this convex continuous loss can be linearized, we end up with linear programming when aggregation functions linear in their parameters are considered. An extension of DHM with surrogate loss that preserves its theoretical properties has been proposed [DeSalvo et al., 2021] and could then be used in this case.

Chapter 6
Online Learning of Capacity-based Preference Models

Contents

Summary

In this chapter, we introduce an *online algorithm* for learning *capacity-based preference models*, designed for decision contexts where preference examples arrive sequentially. Specifically, the proposed method learns *sparse* representations of capacities by leveraging *regularized dual averaging* with ℓ_1-regularization, which reduces the online learning problem to simple optimization tasks that admit closed-form solutions. Thus, the proposed algorithm is also well fitted to decision contexts involving a large number of preference examples or a large number of criteria. Moreover, we propose a variant making it possible to include *normative constraints on the capacity* (e.g., monotonicity, supermodularity), based on the *alternating direction method of multipliers*. This chapter is based on the following publication: [Herin et al., 2024d].

Introduction

In this chapter, we propose online algorithms for learning capacity-based preference models (including the Choquet integral and multilinear utility). While the identification of capacities in preference models has already been the subject of several studies [Grabisch et al., 2008, Tehrani and Hüllermeier, 2013, Benabbou et al., 2017a, Beliakov and Wu, 2019a, Bresson et al., 2020, Pelegrina et al., 2020b, Herin et al., 2023a], some of which were presented in previous chapters, the potential contribution of online learning to the identification of capacities remains underexplored despite a recent attempt [Kakula et al., 2020b] focusing on the Choquet integral without the monotonicity constraint. Indeed, the step of learning preferences is often envisaged in batch mode, i.e., it is assumed that a history of previous decisions is available, or a database of examples of pairwise comparisons, which will be exploited in its entirety to adapt a generic decision model to the user's value system [Fürnkranz and Hüllermeier, 2010a, Domshlak et al., 2011a, Aggarwal and Fallah Tehrani, 2019]. However, in other contexts, particularly that of recommender systems [Zhao et al., 2016], examples of preferences arrive sequentially, because they are collected progressively from recent user's feedback or answers to preference queries. In this case, for reasons of reactivity, it is generally preferable to adapt the current model to the margin using the new example (online learning), rather than restart the learning process from scratch on the set of available examples. When the entire database of examples is available but very large, it can also be efficient to consider these examples sequentially and use online learning [Shalev-Shwartz, 2012, Hoi et al., 2021].

Contributions and Chapter Organization Our contribution in this chapter is to introduce *online learning algorithms suitable for a wide class of capacity-based decision models*, including the Choquet integral and the multilinear model. Specifically, we build on the *regularized dual averaging* (RDA) method [Xiao, 2010] to learn *sparse Möbius representations of capacity* with a *computationally efficient* online learning procedure, allowing to handle problems involving up to more than 20 criteria (Section 1). We also proposes a novel extension of RDA to include *normative constraints on capacities* such as monotonicity and supermodularity, by combining RDA and the *alternating direction method of multipliers* (ADMM) [Glowinski and Marroco, 1975, Boyd et al., 2011] (Section 2). Finally, Section 3 presents numerical test results illustrating the effectiveness of the proposed approach.

Notations Recall that N denotes the set of viewpoints, i.e., $N = \{1, \dots, n\}$ and that the notation $S \subseteq N$ excludes the empty set by convention. The set of alternatives is defined as $\mathcal{X} = [0,1]^n$. Also, for any $x = (x_1, \dots, x_n) \in \mathcal{X}$ and $S \subseteq N$, x_S refers to the restriction of x to the components $x_i, i \in S$.

In this chapter, as in Chapter 3, we consider a general *capacity-based preference model* F_m, that associates to any alternative $x \in \mathcal{X}$, the value:

$$F_m(x) = \textstyle\sum_{S \subseteq N} m(S)\phi_S(x_S) \tag{6.1}$$

where for any $S \subseteq N$, $m(S)$ is the Möbius mass on S, and the associated capacity is defined by $w(S) = \sum_{T \subseteq S} m(T)$. Additionally, ϕ_S aggregates the quantities $x_i, i \in S$ to define the *interaction term* $\phi_S(x_S)$. Recall that ϕ_S is the product if F_m is the *multilinear utility* (see Equation 3.1) and ϕ_S is the min (resp. max) operation if F_m is the conjunctive (resp. disjunctive) form of the *Choquet integral* (see Equation 3.2 and 3.3). Note that, from now on, m and $\phi(x)$ respectively refer to the vectors $m = (m(S))_{S \subseteq N}$ and $\phi(x) = (\phi_S(x_S))_{S \subseteq N}$, indexed by the subsets $S \subseteq N$ numbered in lexicographic order. Using these notations, function $F_m(x)$ reads as the following inner product $F_m(x) = m^\top \phi(x)$. The set of admissible vector m is denoted by $\mathcal{M}$, and is typically a subset of $\mathbb{R}^{2^n - 1}$.

Additionnally, for any vector $u \in \mathbb{R}^d$, $[u]_+$ denotes the component-wise positive part, i.e., $[u]_+ = (\max(0, u_i))_{i=1}^d$, and $\text{sign}(u)$ denotes the vector whose components are defined by $\text{sign}(u)_i = 1$ if $u_i > 0$, $\text{sign}(u)_i = -1$ if $u_i < 0$ and $\text{sign}(u)_i = 0$ otherwise, $i = 1, \ldots, d$. Also, for any sequence of vector $\{u_\tau\}_{\tau=1}^t$, $\bar{u}_t$ denotes its average value, i.e., $\bar{u}_t = \frac{1}{t}\sum_{\tau=1}^t u_\tau$, and for two vectors $v, u \in \mathbb{R}^d$, $v * u$ denotes the element-wise product. Finally, for easier reading, a summary of the acronyms used in this chapter is provided:

1 Online Learning of Sparse Möbius Representations of the Capacity

1.1 Online Sparse Learning

1.1.1 Online Learning and Online Convex Optimization

Online learning algorithms [Shalev-Shwartz, 2012, Orabona, 2019, Hoi et al., 2021] work sequentially: starting from an initial model m_1, the algorithm updates it iteratively as new instances are observed. More precisely, at each round t, a new instance is received, the learner makes a prediction using the current model m_t, and the true label of the instance is received. In the case of an incorrect prediction, the learner then suffers a certain loss $l_t(m_t)$, where $l_t : \mathcal{M} \to \mathbb{R}$ is an instantaneous loss function, and updates the model accordingly. For instance, when learning F_m (see Equation 6.1) in a regression setting, the learner typically receives at round t an instance $x_t \in \mathcal{X}$, makes a prediction $F_{m_t}(x_t) = m_t^\top \phi(x_t)$, receives the true value $y_t \in \mathbb{R}$, and then incurs a loss measured as

Acronyms	Description	Reference
OCO	*Online Convex Optimization*	page 200
OGD	*Online Gradient Descent*	page 201
OMD	*Online Mirror Descent* Generalization of OGD with Bergman divergences. One of the two main families of OCO algorithms.	page 202
FOBOS	*Forward-Backward Splitting* OGD with composite loss.	page 203
FTRL	*Follow-the-Regularized-Leader* One of the two main families of OCO algorithms.	page 204
RDA	*Regularized Dual Averaging* FTRL with composite loss.	page 204
ADMM	*Alternating Direction Method of Multipliers* Family of optimization algorithms leveraging problem decompositions.	page 211

Table 6.1: List of acronyms used in this chapter.

the discrepancy between the predicted value and y_t, i.e., $l_t(m_t) = l(y_t, m_t^\top \phi(x_t))$, where $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a regression loss function.

In what follows, received loss functions l_t are assumed to be convex functions and $\mathcal{M}$ is a non-empty closed convex set. In this case, the online learning procedure falls into *online convex optimization* (OCO) [Shalev-Shwartz, 2012, Hazan et al., 2016], which may be formalized as in Algorithm 6.1.

Algorithm 6.1: Online Convex Optimization

```
Inputs: a convex set $\mathcal{M}$, a total number of iterations $T$
for $t = 1, \dots, T$ do
    pick $m_t \in \mathcal{M}$
    receive a convex loss $l_t : \mathcal{M} \to \mathbb{R}$
    incur loss $l_t(m_t)$
Outputs: $m_T$
```

One desirable property of an OCO algorithm is the guarantee that the cumulative loss after T rounds is close to the minimal cumulative loss one could obtain with all the instances in hand. The gap between the two quantities is referred to as the *regret against the best fixed model* and can be formally defined as follows:

Definition 6.1. *For any $T \in \mathbb{N}$ and any sequences of losses and models $\{l_t, m_t\}_{t=1,\dots,T}$,* *the regret against the best fixed model, denoted by R_T, is defined by:*

$$R_T = \sum_{t=1}^{T} l_t(m_t) - \min_{m \in \mathcal{M}} \sum_{t=1}^{T} l_t(m)$$

Also, the regret w.r.t. any model $m \in \mathcal{M}$ is denoted by $R_T(m) = \sum_{t=1}^{T} l_t(m_t) - \sum_{t=1}^{T} l_t(m)$.

Thus, an OCO algorithm typically updates model m_t at each round t, in such a way that *the average regret against the best fixed model vanishes when the number of rounds goes towards infinity*, i.e., $\lim_{T\to\infty} \frac{R_T}{T} = 0$. The regret is then said to be *sublinear* in T. As we will see in what follows, guaranteeing such behavior is possible under the following assumption, which we will consider to hold throughout the chapter.

Assumption 6.1 (bounded loss subgradients). *For any t, function l_t is subdifferentiable and of bounded subgradients, i.e., there exists $G \in \mathbb{R}_+$ such that, $\|g_t\|_2 \leq G$ for any $g_t \in \partial l_t(m_t)$, $m_t \in \mathcal{M}$.*

Note that in the context of learning the model F_m, Assumption 6.1 may require considering a bounded set $\mathcal{X}$, and in some cases, even a bounded model space $\mathcal{M}$ and a bounded set of labels, as for instance, in the regression setting with the squared loss $l_t(m_t) = (y_t - m_t^\top \phi(x_t))^2$, we have $\partial l_t(m_t) = \{-2\phi(x_t)(y_t - m_t^\top \phi(x_t))\}$.

A simple example of OCO algorithm is Online (sub)Gradient Descent (OGD) that uses the update $m_{t+1} = \Pi_{\mathcal{M}}(m_t - \eta_t g_t)$ where $g_t \in \partial l_t(m_t)$, $\eta_t \in \mathbb{R}_+$ is a learning rate and $\Pi_{\mathcal{M}}$ is the Euclidean projection on $\mathcal{M}$ (i.e, $\Pi_{\mathcal{M}}(m_0) = \arg\min_{m\in\mathcal{M}} \|m - m_0\|_2^2$). If $\eta_t = \Theta(1/\sqrt{t})$ and $\mathcal{M}$ is a bounded set, OGD is known to achieve sublinear regret with a $O(\sqrt{T})$ upper bound [Zinkevich, 2003]. To aid understanding, we provide in the following a sketch of the proof for the case where $\mathcal{M}$ is such that $\max_{m,w\in\mathcal{M}} \|m - w\|_2^2 = D^2$, and $\eta_t = \frac{1}{\sqrt{t}}$. First, by the convexity of l_t and the definition of subgradients (see Definition **??**), we have for any $m \in \mathcal{M}$:

$$R_T(m) = \sum_{t=1}^{T} l_t(m_t) - \sum_{t=1}^{T} l_t(m) \leq \sum_{t=1}^{T} g_t^\top (m_t - m)$$

Moreover, for any t, we have: $\|m_t - \eta_t g_t - m\|_2^2 = \|m_t - m\|_2^2 - 2\eta_t g_t^\top (m_t - m) + \eta_t^2 \|g_t\|_2^2$. Then, using $\|z - m\|_2^2 \geq \|\Pi_{\mathcal{M}}(z) - m\|_2^2$ for any $z \in \mathbb{R}^d, m \in \mathcal{M}$ (see for instance Proposition 2.11 in [Orabona, 2019]), we obtain:

$$g_t^\top (m_t - m) \leq \frac{1}{2\eta_t}(\|m_t - m\|_2^2 - \|m_{t+1} - m\|_2^2) + \frac{\eta_t}{2}\|g_t\|_2^2 \qquad (6.2)$$

Then, summing Equation 6.2 over $t = 1, \dots, T$ and using Assumption 6.1, we have:

$$R_T(m) \leq \frac{1}{2\eta_1}\|m_1 - m\|_2^2 - \frac{1}{2\eta_T}\|m_{T+1} - m\|_2^2 + \frac{1}{2}\sum_{t=2}^{T}\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right)\|m_t - m\|_2^2 + \frac{G^2}{2}\sum_{t=1}^{T}\eta_t$$

Finally, using $\|m_{T+1} - m\|_2^2 \geq 0$, $\max_{m,w\in\mathcal{M}}\|m - w\|_2^2 = D^2$, and $\eta_t = \frac{1}{\sqrt{t}}$, we obtain:

$$\begin{aligned}
R_T(m) &\leq D^2\left(\frac{1}{2\eta_1} + \frac{1}{2}\sum_{t=2}^{T}\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right)\right) + \frac{G^2}{2}\sum_{t=1}^{T}\eta_t \\
&\leq D^2\frac{1}{2\eta_T} + \frac{G^2}{2}\sum_{t=1}^{T}\eta_t \\
&\leq D^2\frac{\sqrt{T}}{2} + \frac{G^2}{2}(2\sqrt{T} - 1)
\end{aligned}$$

where in the last line we used $\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \leq 2\sqrt{T} - 1$ ([Zinkevich, 2003]). Despite its simplicity, OGD achieves the lowest possible upper bound on the regret. Indeed, it is known that we can exhibit a sequence of losses $l_1, \dots, l_T$ such that $R_T \geq \frac{GD\sqrt{T}}{2}$ whatever the algorithm (see [Orabona, 2019] (Theorem 5.1) or [Hazan et al., 2016] (Theorem 3.2)).

OGD falls under a more general family of algorithms that are the *online mirror descent* (OMD) algorithms [Beck and Teboulle, 2003] characterized by the following model update:

$$m_{t+1} = \arg\min_{m\in\mathcal{M}}\left\{g_t^\top m + \frac{1}{\eta_t}B_\psi(m, m_t)\right\} \tag{6.3}$$

where B_ψ is a *Bregman divergence* [Bregman, 1967], i.e., a function of the form $B_\psi(m, m') = \psi(m) - \psi(m') - \nabla\psi(m')^T(m - m')$, where ψ is a differentiable and strictly convex function (see Definition 1.24). By the strict convexity of ψ, for a fix reference point $m' \in \mathcal{M}$, $B_\psi(m, m') \geq 0$ for any $m \in \mathcal{M}$, with equality if and only if $m = m'$. Therefore, B_ψ can be interpreted as a similarity measure, and *Problem 6.3 amounts to minimizing a linearized version of the loss around m_t while staying close to m_t, achieving a form of exploration-exploitation trade-off.*

For instance, by taking $\psi(m) = \frac{1}{2}\|m\|_2^2$, i.e., $B_\psi(m, m_t) = \frac{1}{2}\|m_t - m\|_2^2$, OMD boils down to OGD as in this case Problem 6.3 reduces to:

$$\begin{aligned}
m_{t+1} &= \arg\min_{m\in\mathcal{M}}\left\{g_t^\top m + \frac{1}{2\eta_t}\|m_t - m\|_2^2\right\} \qquad (6.4)\\
&= \arg\min_{m\in\mathcal{M}}\left\{\frac{1}{2}\|m_t - \eta_t g_t - m\|_2^2\right\} \\
&= \Pi_{\mathcal{M}}(m_t - \eta_t g_t)
\end{aligned}$$

In the general case, when $\eta_t = \Theta(1/\sqrt{t})$ and $\mathcal{M}$ is a bounded set, OMD is also known to achieve a $O(\sqrt{T})$ regret upper bound [Warmuth et al., 1997, Beck and Teboulle, 2003, Cesa-Bianchi and Lugosi, 2006].

Remark 6.1 (online and stochastic learning). It is important to note that, in the online learning literature, the aim is to design algorithm that guarantee a $O(\sqrt{T})$ regret bound, without making any assumptions about the sequence of losses received, which may be stochastic, deterministic or even adversarial (arbitrarily chosen by an adversary). For this reason, the learning process is commonly interpreted as a *game* in which the learner picks a decision m_t, and the adversary determines a loss $l_t(m_t)$ [Cesa-Bianchi and Lugosi, 2006, Shalev-Shwartz, 2012]. Thus, this setting is more general than the related one of *stochastic learning* [Robbins and Monro, 1951, Polyak and Juditsky, 1992, Bottou and Cun, 2003, Bottou et al., 2018], where the learning problem is also addressed through sequential processing of instances. In the latter setting, the losses are indeed assumed to take the form $l_t(m_t) = l(m_t^\top \phi(x_t), y_t)$, where the stream of instances (x_t, y_t) is i.i.d. from an unknown but fixed distribution. For a discussion on the links between both settings, the interested reader may refer to Shalev-Shwartz [2012] (Chapter 5) and [Orabona, 2019] (Chapter 3).

1.1.2 Online Learning with ℓ_1-regularization

Let us now consider a ℓ_1-regularized loss, i.e., for any t, and any $m \in \mathcal{M}$:

$$f_t(m) = l_t(m) + \lambda \|m\|_1 \tag{6.5}$$

where $\lambda > 0$ is a hyperparameter that controls the level of regularization. Note that functions $f_t, t = 1, \dots, T$ are convex functions.

Including ℓ_1-regularization in the loss functions requires special care, as applying an OMD algorithm directly to f_t would amount linearizing both l_t and the ℓ_1-norm, i.e., computing m_{t+1} via Equation 6.3 with $g_t \in \partial f(m_t)$, thereby cancelling out the desired sparsity effect of the ℓ_1-regularization. Thus, OMD algorithms have been extended to handle ℓ_1-regularized losses or other *composite loss* functions of the form $l_t(m) + r(m)$, where r is typically a regularization function [Langford et al., 2009, Duchi and Singer, 2009, Duchi et al., 2010].

For instance, the FOBOS algorithm [Duchi and Singer, 2009] is a variant of OGD (see Equation 6.4) where the ℓ_1-norm regularization is not linearized, i.e.:

$$m_{t+1} = \operatorname*{arg\,min}_{m \in \mathcal{M}} \left\{ g_t^\top m + \lambda \|m\|_1 + \frac{1}{2\eta_t} \|m_t - m\|_2^2 \right\} \tag{6.6}$$

Each step of FOBOS solves an ℓ_1-regularized optimization problem, enabling the recovery of sparse models. Furthermore, this problem admits a closed-form solution when $\mathcal{M} = \mathbb{R}^d$, which can derived using the notion of *proximal operator*, that we define below:

Definition 6.2 (proximal operator). *The proximal operator of a proper and convex function $h : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is defined as:*

$$\operatorname{prox}_h(x) = \arg\min_{z \in \mathbb{R}^d} \left(\frac{1}{2} \|z - x\|_2^2 + h(z) \right) \tag{6.7}$$

Then, when $h(x) = \lambda \|x\|_1$, prox_h admits the following analytical form (the proof of this standard result is given for completeness in Lemma 2 of Appendix C):

$$\operatorname{prox}_{\lambda\|\cdot\|_1}(x) = \operatorname{sign}(x) * [|x| - \lambda]_+ \tag{6.8}$$

Therefore, using Equation 6.8, it can easily be checked that Problem 6.6 admits the following closed-form solution when $\mathcal{M} = \mathbb{R}^d$:

$$m_{t+1} = \operatorname{sign}(m_t - \eta_t g_t) * [|m_t - \eta_t g_t| - \lambda \eta_t]_+$$

However, FOBOS has shown difficulties in fully exploiting the ℓ_1-regularization and in particular provides models with high numbers of non-null coefficients [Xiao, 2009]. Nevertheless, another major family of online algorithms, called *follow-the-regularized-leader* (FTRL) [Shalev-Shwartz, 2007, 2012], when combined with ℓ_1-regularized losses, is known to produce enhanced sparse models compared to OMD methods (such as FOBOS) [Xiao, 2009, McMahan, 2011, Hoi et al., 2021]. The resulting algorithms appear under the name *regularized dual averaging* (RDA)[Xiao, 2009, 2010]. FTRL and RDA are presented in the following section.

1.1.3 Follow-the-regularized-leader (FTRL)

FTRL algorithms consist in taking m_{t+1} as the model that minimizes the average loss received on the past rounds and some regularization function $\psi : \mathcal{M} \to \mathbb{R}$, i.e.,:

$$m_{t+1} = \arg\min_{m \in \mathcal{M}} \left\{ \sum_{\tau=1}^{t} l_\tau(m) + \frac{1}{\eta_t} \psi(m) \right\} \tag{6.9}$$

Thus, contrarily to OMD that computes m_{t+1} based on m_t and the loss received at step t (see Problem 6.3), *FTRL uses the whole history of received losses and computes m_{t+1} as the model that, while remaining simple, would have perform best over the past rounds (the leader).* When ψ is closed and strongly convex (see Definition 1.26 and 1.25) and $\eta_t = \Theta(1/\sqrt{t})$, such scheme is known to admit a $O(\sqrt{T})$ regret bound [Shalev-

Shwartz, 2007, Xiao, 2010, Shalev-Shwartz, 2012, Orabona, 2019]. The detailed result is provided below:

Theorem 6.1. *Let $\psi : \mathcal{M} \to \mathbb{R}$ be a closed and 1-strongly convex function, and such that $\min_{m \in \mathcal{M}} \psi(m) = 0$. Let $\eta_t = \frac{\gamma}{\sqrt{t}}$ and $\mathcal{F}_D = \{m \in \mathcal{M} : \psi(m) \leq D^2\}$ for some $D, \gamma \in \mathbb{R}_+^*$. Then, under Assumption 6.1, for any $m \in \mathcal{F}_D$:*

$$R_T(m) \leq (\frac{D^2}{\gamma} + \gamma G^2)\sqrt{T} \tag{6.10}$$

A proof can be found in [Xiao, 2010] (Corollary 2), and in [Orabona, 2019] (Corollary 7.9) for a factor η_t ahead of one time-step, i.e., $\eta_t = \frac{\gamma}{\sqrt{t+1}}$, or in [Shalev-Shwartz, 2012] (Theorem 2.11) for a non-time varying factor, i.e., $\eta_t = \eta$.

As it is, the FTRL update given by Equation 6.9, requires solving an optimization problem at each iteration. However, similarly as in OMD, it can be alleviated by considering the linearized loss $\tilde{l}_t(m) = g_t^\top m$, $g_t \in \partial l_t(m_t)$. Indeed, under Assumption 6.1, as $\partial \tilde{l}_t(m_t) = \{g_t\} \subseteq \partial l_t(m_t)$, losses $\tilde{l}_t$ naturally also satisfy Assumption 6.1 with the same gradient bound G. Therefore, using Theorem 6.1 with $\tilde{l}_t(m_t)$ and the definition of the subgradients (see Definition 1.27), we have for any $m \in \mathcal{F}_D$:

$$\sum_{t=1}^{T}(l_t(m_t) - l_t(m)) \leq \sum_{t=1}^{T} g_t^\top (m_t - m) = \sum_{t=1}^{T}(\tilde{l}_t(m_t) - \tilde{l}_t(m)) \leq (\frac{D^2}{\gamma} + \gamma G^2)\sqrt{T}$$

Therefore, considering l_t or its linearized version $\tilde{l}_t$ is equivalent, and using $\tilde{l}_t$ in Problem 6.9 typically gives closed-form solutions. For instance, when $\psi(m) = \frac{1}{2}\|m\|_2^2$, it yields the following update:

$$m_{t+1} = \operatorname*{arg\,min}_{m \in \mathcal{M}} \left\{ \sum_{\tau=1}^{t} g_t^\top m + \frac{1}{2\eta_t}\|m\|_2^2 \right\} \tag{6.11}$$

$$m_{t+1} = \operatorname*{arg\,min}_{m \in \mathcal{M}} \left\{ \bar{g}_t^\top m + \frac{1}{2\eta_t t}\|m\|_2^2 \right\} \tag{6.12}$$

$$m_{t+1} = \operatorname*{arg\,min}_{m \in \mathcal{M}} \left\{ \frac{1}{2\eta_t t}\|m - (-\eta_t t \bar{g}_t)\|_2^2 \right\} \tag{6.13}$$

$$= \Pi_{\mathcal{M}}\left(-\eta_t t \bar{g}_t\right) \tag{6.14}$$

1.1.4 Regularized Dual Averaging (RDA): FTRL with Composite Loss

FTRL with a composite loss of the form $l_t(m) + r(m)$ yields the following update:

$$m_{t+1} = \arg\min_{m\in\mathcal{M}} \Big\{\frac{1}{t}\sum_{\tau=1}^{t} l_\tau(m) + r(m) + \frac{1}{t\eta_t}\psi(m)\Big\} \tag{6.15}$$

where the objective function has been divided by t as in Equation 6.12. Such update is known under the name *regularized dual averaging* (RDA) [Xiao, 2010] (a name that comes from the optimization literature as RDA can be seen as an extension of the *dual averaging* method [Nesterov, 2009]).

For $r(m) = \lambda\|m\|_1$, RDA has been empirically shown to produce sparser models compared to OMD methods like FOBOS [Xiao, 2009, McMahan, 2011, Hoi et al., 2021]. This is further supported by theoretical results that show that, while FOBOS does apply a ℓ_1-regularization at round t (see Equation 6.6), it amounts to using a linearized version of the ℓ_1-norm for all previous rounds (when written in a FTLR form) [McMahan, 2011]. In contrast, the RDA update (Equation 6.15) involves applying an explicit ℓ_1 regularization to all past rounds.

The sparsity effect of this ℓ_1-regularization can be preserved while exploiting the benefit of loss linearization for computational efficiency by solely linearizing l_t, i.e., using $\tilde{l}_t(m)$. In this case, using Equation 6.8, it can easily be checked that Problem 6.15 admits the following closed-form solution when $r(m) = \lambda\|m\|_1$ and $\mathcal{M} = \mathbb{R}^d$:

$$m_{t+1} = -t\eta_t\Big[|\bar{g}_t| - \lambda\Big]_+ * \operatorname{sign}(\bar{g}_t) \tag{6.16}$$

Remark 6.2 (regret bound with composite loss). To obtain a regret bound, Theorem 6.1 can be applied to the partially linearized composite loss, i.e., $\tilde{l}_t(m)+\lambda\|m\|_1$. However, the obtained bound now is of the form $(\frac{D^2}{\gamma} + \gamma(G^2+\lambda^2 d))\sqrt{T}$ as it can easily be checked that for any $s \in \partial\|.\|_1(m_t)$, $\|s\|_2^2 \leq d$. Several proposed regret analysis allow bypassing this issue and provide a proof of a bound depending on G solely (see Xiao [2010] (Corollary 2) or Orabona [2019] (Section 7.8)), guaranteeing that for any $m \in \mathcal{F}_D$:

$$\sum_{t=1}^{T}(\tilde{l}_t(m_t) + \lambda\|m_t\|_1) - \sum_{t=1}^{T}(\tilde{l}_t(m) + \lambda\|m\|_1) \leq (\frac{D^2}{\gamma} + \gamma G^2)\sqrt{T}$$

In the following, we provide the explicit RDA algorithm for learning sparse Möbius representation of the capacity. Additionally, the benefit of using an RDA algorithm over FOBOS is illustrated with numerical experiments in Section 3.

1.2 A RDA Algorithm for Preference Learning

We now consider the online setting to learn a *sparse* Möbius vector m in model F_m (see Equation 6.1) from *preference examples.* The preference examples are supposed to be received as a stream of pairwise preference examples $(x_t, x'_t) \in \mathcal{X}^2$, where at each round t, we consider without loss of generality that $x_t \succ x'_t$ (strict preference) or $x_t \sim x'_t$ (indifference). In this setting, a natural convex loss function is $l_t(m) = l(m^\top \phi(x_t), m^\top \phi(x'_t))$ where l is the *pref-hinge* loss (see Definition 1.28), i.e.:

$$\begin{aligned} l_t(m) &= [\delta - m^\top(\phi(x_t) - \phi(x'_t))]_+ && \text{if } t \in P \\ &= [|m^\top(\phi(x_t) - \phi(x'_t))| - \delta]_+ && \text{if } t \in I \end{aligned} \tag{6.17}$$

where P (resp. I) denotes the index set of preference (resp. indifference) examples and $\delta > 0$ is a discrimination threshold used to separate preference from indifference situations. Thus l_t measures the violation of preference $x_t \succ x'_t$ if $t \in P$ or indifference $x_t \sim x'_t$ if $t \in I$. Here, to promote sparse solutions and thus obtain sparse Möbius representations of capacities, we use the ℓ_1-regularized version of the loss (see Equation 6.5).

By definition of l_t (see Equation 6.17), it can easily be checked that, for any $m \in \mathcal{M}$, the following vector belongs to $\partial l_t(m)$:

$$\begin{aligned} g_t &= (\phi(x'_t) - \phi(x_t))\,\mathrm{sign}(l_t(m)) && \text{if } t \in P \\ &= (\phi(x'_t) - \phi(x_t))\,\mathrm{sign}(l_t(m_t)m^\top(\phi(x'_t) - \phi(x_t)) && \text{if } t \in I \end{aligned} \tag{6.18}$$

Remark that Assumption 6.1 holds for any continuous function ϕ and compact (closed and bounded) set $\mathcal{X}$, as for any $g_t \in \partial l_t(m)$, we have $\|g_t\|_2^2 \leq \|\phi(x'_t) - \phi(x_t)\|_2^2 \leq \max_{x,x' \in \mathcal{X}^2} \|\phi(x) - \phi(x')\|_2^2$. Then, the online learning algorithm based on Equation (6.16) for learning a compact Möbius representation of capacities in $\mathcal{M} = \mathbb{R}^d$ (with $d = 2^n - 1$) is summarized in Algorithm 6.2 for $\eta_t = \frac{\gamma}{\sqrt{t}}$ where Equation 6.16 corresponds to line 8.

Algorithm 6.2: RDA for Preference Learning

```
   Inputs: $(\gamma, \lambda, T)$
1  $t \leftarrow 1$, $m_1 \leftarrow (0, \ldots, 0)$, $\overline{g}_0 \leftarrow 0$
2  while $t < T$ do
3      receive pairwise example $(x_t, x'_t)$
4      compute loss gradient $g_t \in \partial l_t(m_t)$ according to Equation 6.18
5      # update average gradient
6      $\bar{g}_t \leftarrow \frac{t-1}{t} g_{t-1} + \frac{1}{t} g_t$
7      # update model
8      $m_{t+1} \leftarrow -\gamma\sqrt{t}\Big[|\bar{g}_t| - \lambda\Big]_+ * \mathrm{sign}(\bar{g}_t)$
9      $t \leftarrow t + 1$
   Outputs: $m_T$
```

Associated batch learning problem It is important to note that the learning problem with all instance in hands, i.e., a training set of pairwise comparisons $\{x_t, x'_t\}_{t=1}^{T}$ with $T = |P| + |I|$, corresponds to the following *batch* learning problem:

$$\begin{aligned}\min_{m\in\mathcal{M}} \frac{1}{T}(\sum_{t\in P}\epsilon_t + \sum_{t\in I}(\epsilon_t^- + \epsilon_t^+)) + \lambda\|m\|_1 \qquad (6.19)\\ m^\top(\phi(x_t) - \phi(x'_t)) \geq \delta - \epsilon_t,\ t \in P\\ m^\top(\phi(x_t) - \phi(x'_t)) \leq \delta + \epsilon_t^+,\ t \in I\\ m^\top(\phi(x'_t) - \phi(x_t)) \leq \delta + \epsilon_t^-,\ t \in I\\ \epsilon_t, \epsilon_t^+, \epsilon_t^- \geq 0,\quad t = 1, \ldots, T\end{aligned}$$

where variables ϵ_t (resp. ϵ_t^+,ϵ_t^+) are variables modeling the error on preference (resp. indifference) examples introduced to linearize loss l_t (see Remark 2.2).

Thus, evaluating the performance of an online algorithm using the average regret against the best fixed model (see Definition 6.1) amounts to comparing, in hindsight, the average loss incurred by the online algorithm with the optimal value of Problem 6.19. Remark that this batch problem coincides with the learning problem studied in Chapter 3 (see Problem 3.8), and that it can be solved with high precision using linear programming up to a dozen of criteria. This solving method is used to compute the average regret associated with Algorithm 6.2 in the numerical experiments of Section 3.

Also, it should be emphasized that the proposed online approach has the well-known advantage of *improving the scalability of the learning task*, compared with batch problem solving (6.19). Due to the efficient closed-form of Equation (6.16), *Algorithm 6.2 applies on instances involving more than 20 criteria (millions of possible interactions)*, as shown by the results of numerical tests given in Section 3. Handling problems with such size is also possible with the algorithm proposed in Chapter 3 (see Algorithm 3.2) in batch mode, provided the database of preference examples is small (a few hundreds). Here, the computational complexity of Algorithm 6.2 is in $O(Td)$. It is still exponential in the number of criteria since $d = 2^n - 1$ but linear in T for bounded n. This is an advantage in view of processing large-size databases.

In the next section, we address the challenge of learning a sparse and *constrained* preference model.

2 Online Sparse Learning with Constraints

2.1 Structural Constraints on the Capacity

A key feature of preference model F_m arises when the interaction function is chosen as the functions $\phi_S(x_S) = \min_{i\in S}\{x_i\}$ or $\phi_S(x_S) = \prod_{i\in S} x_i$ (respectively yielding the Choquet integral and the multilinear model), and m is associated to a *monotonic* capacity w, i.e., such that for any $T \subseteq S$, $w(S) \leq w(T)$ ($\Leftrightarrow \sum_{S'\subseteq S} m(S') \leq \sum_{T'\subseteq T} m(T)$). Indeed, under these conditions, F_m proves to be *monotonic*, i.e., $\forall i \in N, x_i \geq x'_i \Rightarrow F_m(x) \geq F_m(x')$ (see [Grabisch, 2016a], Chapter 4), making the preferences induced by F_m consistent with weak Pareto dominance. Thus, when learning the capacity from preference examples, whether in the Choquet integral, in the multilinear model, or more generally in the F_m model, the question arises as to how to obtain a capacity that verifies this monotonicity property.

Let us first remark that preference examples may partly contribute to enforce monotonicity. For instance, in the case of Choquet and multilinear model we have $F_m(1_S, 0_{-S}) = w(S)$ for all $S \subseteq N$ where $(1_S, 0_{-S})$ is the vector of $\mathbb{R}^n$ whose components indexed in S equal 1, the other being equal to 0. Hence, for any pair T, S of subsets such that $T \subseteq S \subseteq N$, a preference example like $(1_S, 0_{-S}) \succsim (1_T, 0_{-T})$ is equivalent to $w(S) \geq w(T)$. Thus a capacity-based decision model that well fits such preference examples should nearly satisfy monotonicity on the pairs present in the database. However, in practice, preference are collected from past experiences and we cannot expect that all relevant (S, T) pairs are present in the preference database. Multiple violations of monotonicity are still possible. Another approach to enforce monotonicity is to explicitly include all monotonicity constraints in the learning process. Note that the constraints expressed using Möbius masses reduces as follows: $\sum_{T\subseteq S, T\ni i} m(T) \geq 0, \quad \forall i \in S, \forall S \subseteq N$. This approach was used in the batch setting in Chapter 2 (see Problem **??**) and combined with constraints generation in Chapter 3 (see Algorithm 3.4). In this section, this option is investigated in the online setting.

Beyond monotonicity, other structural constraints on the capacity might be considered, and in particular *supermodularity*. A capacity w is said to be *supermodular* (or convex) if $w(S \cup T) + w(S \cap T) \geq w(S) + w(T)$ for all $S, T \subseteq N$. This condition is used in the Choquet integral to support the emergence of fair solutions in choices [Lesca and Perny, 2010]. More precisely, if the decision maker is indifferent between q solutions $x^1, \ldots, x^q$, with a supermodular w it is guaranteed that a vector obtained by convex combination of $x^1, \ldots, x^q$ will be preferred to any of the x^i's [Chateauneuf and Tallon, 2002a]. Thus, softening the variations of components in vectors $x^i, i = 1, \ldots, q$ makes the decision maker better off. This condition promotes alternatives having balanced profiles. It is illustrated on a toy example below.

Example 6.1. *If the decision maker is indifferent between (1, 0) and (0, 1), a solution like* $(\frac{1}{2}, \frac{1}{2})$ *will be preferred to the other two according to the Choquet model, provided that the capacity is supermodular. We have indeed, if* m *is the Möbius transformed of the capacity* w*,* $F_m(1,0) = w(\{1\})$*,* $F_m(0,1) = w(\{2\})$ *and* $w(\{1\}) = w(\{2\})$ *since* $(1,0)$ *and* $(0,1)$ *are indifferent. Hence* $F_m(\frac{1}{2}, \frac{1}{2}) = \frac{1}{2}w(\{1,2\}) \geq \frac{1}{2}(w(\{1\}) + w(\{2\}))$ *by supermodularity of* w*. Therefore* $F_m(\frac{1}{2}, \frac{1}{2}) \geq w(\{1\}) = F_m(1,0) = F_m(0,1)$*.*

Obviously, supermodularity can also be expressed using Möbius masses, and both supermodularity and monotonicity constraints consist in linear constraints in m. Let c denote the number of constraints and $C \in \mathbb{R}^{c \times d}$ the matrix encoding the linear constraints on the capacity such as monotonicity and/or supermodularity constraints, i.e., such that the set of admissible models is $\mathcal{M} = \{m : Cm \leq 0\}$.

Example 6.2. *When* $N = \{1,2\}$ *monotonicity and supermodularity are enforced by the system* $Cm \leq 0$ *with:*

$$C = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & -1 & -1 \\ -1 & 0 & -1 \\ 0 & 0 & -1 \end{pmatrix} \quad \textit{and} \quad m = \begin{pmatrix} m_1 \\ m_2 \\ m_{12} \end{pmatrix}$$

Monotonicity is guaranteed by the four first lines, and supermodularity by the fifth.

The size of matrix C is exponential in n. However, C gets sparser as n increases which allows us to resort to specialized libraries (e.g., scipy.sparse) for efficient matrix products in learning algorithms.

In its current form, Algorithm 6.2 does not enforce monotonicity constraints on the capacity. As suggested previously, if the DM preferences are monotonic w.r.t. Pareto dominance, we may observe in practice that the algorithm progressively captures the data monotonicity as new preference examples arrive (this is confirmed by our numerical tests, see Section 3). However, even though the average monotonicity violation progressively vanishes, high-amplitude and recurrent violations can occur, especially at the beginning of the online learning process. For this reason, in the next section, we propose an extension of Algorithm 6.2 that explicitly includes monotonicity constraints and possibly other constraints such as supermodularity constraints.

2.2 A RDA-ADMM Algorithm

2.2.1 Online Learning with Constraints

A natural way to include the monotonicity and/or modularity constraints in Algorithm 6.2 is to simply consider $\mathcal{M} = \{m : Cm \leq 0\}$. However, in this case the RDA update given by Equation 6.15 with partially linearized loss $\tilde{l}_t(m) + \|m\|_1$ and $\psi(m) = \frac{1}{2}\|m\|_2^2$ no longer admits a closed-form solution (in contrast to the unconstrained case in which the closed-form solution is given by Equation 6.16). Therefore, including constraints requires running a potentially costly optimization procedure at each update.

Another option is to not enforce the constraints at every step but use the concept of *long-term constraints* and guarantee a bound on the cumulative constraint violation, similarly to the regret bound [Mahdavi et al., 2012, Wang and Banerjee, 2012, Jenatton et al., 2016, Yu and Neely, 2020]. Among the long-term constraints methods, online *alternate direction method of multiplier* (ADMM) [Wang and Banerjee, 2012, Suzuki, 2013, Ouyang et al., 2013, Suzuki, 2014, Hosseini et al., 2014, Liu et al., 2018] combine online algorithms with ADMM, a well-known iterative optimization method for batch problems that uses splitting variables to reduce the optimization problem into easier sub-problems at each iteration [Boyd et al., 2011].

Online ADMM algorithms have been proposed to extend both OMD [Wang and Banerjee, 2012, Ouyang et al., 2013, Suzuki, 2013, Liu et al., 2018] and FTRL (RDA) [Suzuki, 2013, Hosseini et al., 2014] to include linear constraints $Am \leq b$ or *structured regularization*, i.e., regularization of the form $r(Am)$. Before exploring these extensions, we first introduce the ADMM optimization method.

Remark 6.3 (constraints without ℓ_1-regularization). It is important to recall that without the ℓ_1-regularization, the RDA update given by Equation 6.15 with linearized loss $\tilde{l}_t(m)$ and $\psi(m) = \frac{1}{2}\|m\|_2^2$ reduces to $m_{t+1} = \Pi_{\mathcal{M}}(-\eta_t t \bar{g}_t)$ as given by Equation ??. Similarly, without ℓ_1-regularization, the OGD update is defined by $m_{t+1} = \Pi_{\mathcal{M}}(m_t - \eta_t g_t)$ (given in page 201). While appearing simple, these updates still involve potentially highly costly projection steps to take back the updated model into the admissible set $\mathcal{M}$ at each iteration. For this reason, projection-free online algorithms have been developed [Hazan and Kale, 2012, Chen et al., 2018, Hazan and Minasyan, 2020]. For instance, an algorithm [Hazan and Kale, 2012] based on the Frank-Wolfe method [Frank et al., 1956] replaces the projection step with linear programming (LP). However, this approach is not very efficient to achieve monotonicity, as LP on the set $\mathcal{M} = \{m : Cm \leq 0\}$ is not known to reduce to a computationally simple problem.

2.2.2 The Alternate Direction Method of Multiplier (ADMM)

ADMM [Glowinski and Marroco, 1975, Gabay and Mercier, 1976, Boyd et al., 2011, Nishihara et al., 2015] is a widely used optimization method for solving large-scale or distributed optimization problems [Forero et al., 2010, Bioucas-Dias and Figueiredo, 2010, Wang et al., 2019, Liu et al., 2020], due to its ability to decompose complex problems into simpler subproblems. Below, we first introduce its precursor, the *method of multipliers* [Powell, 1969].

The Method of Multipliers Let us consider the following optimization problem:

$$\min_x f(x) \qquad (6.20)$$

$$Dx = b$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is convex and $b \in \mathbb{R}^p, D \in \mathbb{R}^{p \times d}$, for any $p, d \in \mathbb{N}$. We now define the *augmented Lagrangian* as follows:

$$\mathcal{L}_\rho(x, \mu) = f(x) - \mu^T(Dx - b) + (\rho/2)\|Dx - b\|_2^2 \qquad (6.21)$$

where $\rho \in \mathbb{R}_+$ and $(x, \mu) \in \mathbb{R}^d \times \mathbb{R}^p$.

Remark that $\mathcal{L}_\rho$ corresponds to the (standard) Lagrangian (see Section 2.2.1 in Chapter 3) of the problem of minimizing $f(x) + \frac{\rho}{2}\|Dx - b\|_2^2$ under the constraint $Dx - b$ (which is equivalent to Problem 6.20). The dual function of the latter problem expresses as $g(\mu) = \min_x \mathcal{L}_\rho(x, \mu)$, which can be shown to be differentiable under mild condition [Boyd et al., 2011]. In this case, its gradient is $\nabla g(\mu) = \mathcal{L}_\rho(x^*, \mu) = Dx^* - b$ where $x^* \in \arg\min \mathcal{L}_\rho(x^*, \mu)$. Then, the Method of Multipliers consists in solving the optimization problem with gradient ascent using ρ as the learning rate, i.e.,:

$$\mu^{k+1} = \mu^k + \rho \nabla g(\mu^k) \Leftrightarrow \begin{cases} x^{k+1} & = \arg\min_x \mathcal{L}_\rho\left(x, \mu^k\right) \\ \mu^{k+1} & = \mu^k + \rho\left(Dx^{k+1} - b\right) \end{cases}$$

A detailed convergence analysis of this iterative optimization procedure can be found in [Bertsekas, 1982] (Chapter 2). This scheme allows replacing a constrained optimization problem with a sequence of unconstrained optimization problems. However, the term $\|Dx - b\|_2^2$ in $\mathcal{L}_\rho$ introduces non-separability, preventing us from exploiting the potential separability of Problem 6.20 (which occurs when f decomposes additively onto a partition of the components of x). The ADMM method, that we introduce below, allows us to bypass this issue.

The Alternating Direction Method of Multipliers (ADMM) For simplicity, let us consider a two-block variable vector $x = (m, z)$ and an objective function that additively decomposes onto this variable separation, i.e., $f(m) = h(m) + u(z)$. Then, if $D = (A, B)$, Problem 6.20 can be written as follows:

$$\begin{aligned} &\min_{m,z} h(m) + u(z) \\ &Am + Bz = b \end{aligned} \tag{6.22}$$

Then, the ADMM consists of the following iterative optimization procedure:

$$m^{k+1} = \arg\min_{m} \mathcal{L}_\rho \left(m, z^k, \mu^k\right) \tag{6.23}$$

$$z^{k+1} = \arg\min_{z} \mathcal{L}_\rho \left(m^{k+1}, z, \mu^k\right) \tag{6.24}$$

$$\mu^{k+1} = \mu^k + \rho \left(Am^{k+1} + Bz^{k+1} - b\right) \tag{6.25}$$

where $\mathcal{L}_\rho$ is the augmented Lagrangian of Problem 6.22.

Unlike the multiplier method, the augmented Lagrangian is now *alternatively* minimizated with respect to m and z at each iteration (thus justifying the name of the method). This results in an iterative process where each sub-problem may be simpler than a joint minimization of the augmented Lagragian. Under the light assumption that Problem 6.22 admits a solution, and h, u are closed proper convex functions (see Definitions 1.24 and 1.26), the ADMM is known to have a $O(1/k)$ convergence rate[1] [He and Yuan, 2012, Wang and Banerjee, 2012]. For a more in-depth introduction to ADMM, the interested reader may refer to Boyd et al. [2011].

In the following, we present ADMM in the online setting, using the RDA algorithm (see Section 1.1.4).

2.2.3 RDA-ADMM

Let us now consider RDA (see Problem 6.15) with $r(m) = u(Am - b)$ with $g : \mathbb{R}^p \to \mathbb{R}$, i.e.,:

$$m_{t+1} = \arg\min_{m \in \mathcal{M}} \left\{ \frac{1}{t} \sum_{\tau=1}^{t} l_\tau(m) + u(Am - b) + \frac{1}{t\eta_t} \psi(m) \right\} \tag{6.26}$$

Such update includes the following two interesting sub-cases:

(1) for $b = 0$ and $u(z) = \lambda \|z\|_1$, Problem 6.27 reduces to a RDA update with *structured* ℓ_1-regularization, i.e., $r(m) = \lambda \|Am\|_1$, a penalization used in the *generalized lasso*

[1] i.e., there exists $C \in \mathbb{R}_+$ such that $h(\overline{m}^k) + u(\overline{z}^k) - (h(m^*) + u(x^*)) \leq \frac{C}{k}$ for any k, where (m^*, z^*) is the optimal solution of Problem 6.22.

[Roth, 2004, Ali and Tibshirani, 2019]. More generally, structured regularization of the form $r(Am)$ can be used to perform *group regularization* [Qin and Goldfarb, 2012, Suzuki, 2013].

(2) for $u(z) = \mathbb{I}_-(z)$ where $\mathbb{I}_-$ is the indicator function of $\mathbb{R}^p_-$, i.e., $\mathbb{I}_-(z) = 0$ if $z \in \mathbb{R}^p_-$ and $+\infty$ otherwise, Problem 6.27 reduces to a FTRL update with linear constraints $Am \leq b$.

In any case, introducing an auxiliary variable $z = Am - b \in \mathcal{Z}$ allows Problem 6.26 to be reformulated as follows:

$$\begin{aligned} (m_{t+1}, z_{t+1}) \in \operatorname*{arg\,min}_{m \in \mathcal{M}, \mathcal{Z}} h_t(m) + u(z) \qquad &\text{with} \quad h_t(m) = \frac{1}{t}\sum_{\tau=1}^{t} l_\tau(m) + \frac{1}{t\eta_t}\psi(m) \\ z = Am - b \qquad & \end{aligned} \tag{6.27}$$

Then, at each round t, the update given by Problem 6.27 is a standard ADMM problem with $B = -I$ (see Problem 6.22), and thus can be solved using ADMM. However, we end up with a two-loop online learning algorithm: for each update, an iterative optimization procedure given by Equations 6.23-6.24 has to be launched. On the other hand, it is known that under some assumptions [Suzuki, 2013], making a single pass of the ADMM iterative procedure on Problem 6.27 allows guaranteeing both a sublinear regret and a sublinear constraint violation. More precisely, if $\mathcal{L}_{t,\rho}$ denotes the augmented Lagrangian of Problem 6.27 for any t, the following iterative procedure has been proposed as an RDA-ADMM algorithm [Suzuki, 2013]:

$$m_{t+1} = \operatorname*{arg\,min}_{m \in \mathcal{M}} \mathcal{L}_{t,\rho}(m, \bar{z}_t, \bar{\mu}_t) \tag{6.28}$$

$$z_{t+1} = \operatorname*{arg\,min}_{z \in \mathcal{Z}} \mathcal{L}_{t,\rho}(m_{t+1}, z, \mu_t) \tag{6.29}$$

$$\mu_{t+1} = \mu_t - \rho(Am_{t+1} - b - z_{t+1}) \tag{6.30}$$

Remark that Equations 6.28-6.30 correspond to a single pass of the ADMM procedure given by Equation 6.23-6.24 applied to Problem 6.27, where in Equation (6.28), z and μ are set to their average values over the past rounds, allowing to keep memory of the past constraint violations [Suzuki, 2013]. The associated regret and constraint violation bound guarantee is provided below.

Theorem 6.2. *(adapted from [Suzuki, 2013]; see Theorem 7 in the paper's appendix) Under Assumption 6.1 and the following additional assumptions:*

(i) $\mathcal{M}$ and $\mathcal{Z}$ are compact convex with radius $D \in \mathbb{R}_+$, i.e., such that $\max_{m,m' \in \mathcal{M}} \|m - m'\|_2 \leq D$ and $\max_{z,z' \in \mathcal{Z}} \|z - z'\|_2 \leq D$,

(ii) $\eta_t = \frac{\gamma}{\sqrt{t}}$ *and* $\psi(m) = \psi_t(m) = \frac{1}{2}(\|m\|_2^2 - \rho\eta_t t\|A(m - \overline{m}_t)\|_2^2)$*, where* $\gamma \in \mathbb{R}_+$ *is taken so that* $\psi_t(m)$ *is a strongly convex regularization for any* t*,*

(iii) u *is subdifferentiable and of bounded subgradients, i.e., there exists* $L \in \mathbb{R}_+$ *such that,* $\|s\|_2 \leq L$ *for any* $s \in \partial u(z)$*,* $z \in \mathcal{Z}$*,*

the update given by Equations 6.28-6.30 yields the following bound:

$$\sum_{t=1}^{T}(l_t(m_t) + u(z_t)) - \sum_{t=1}^{T}(l_t(m) + u(z)) + \frac{\rho}{2}\sum_{t=1}^{T}\|Am_{t+1} + Bz_{t+1} - b\|_2^2 \leq (\frac{D^2}{\gamma} + G^2)\sqrt{T} + K$$

for any $m, z \in \mathcal{M} \times \mathcal{Z}$*, where* K *is a constant depending on* D, G, L, A *and* η_1*.*

Remark 6.4 (Online ADMM with OMD). Similar analysis are also known for the OMD procedure (see Equation 6.3) [Wang and Banerjee, 2012, Suzuki, 2013, Ouyang et al., 2013, Liu et al., 2018]. The aforementioned references constitutes a more extensive body of literature than that on RDA-ADMM, where the only analysis available are, to the best of our knowledge, that of [Suzuki, 2013] and [Hosseini et al., 2014] in the distributed setting.

2.2.4 A RDA-ADMM Algorithm for Sparse and Constrained Online Learning

In order to allow for both a sparse and constrained learning, let us now consider the RDA-ADMM update given by Problem 6.26 with a ℓ_1-regularized loss and $u(z) = \mathbb{I}_{-}(z)$ (setting (2)), i.e.,:

$$m_{t+1}, z_{t+1} = \underset{m \in \mathcal{M}, z \in \mathcal{Z}}{\arg\min} \left\{ \frac{1}{t}\sum_{\tau=1}^{t} l_\tau(m) + \lambda\|m\|_1 + \mathbb{I}_{-}(z) + \frac{1}{t\eta_t}\psi_t(m) \right\} \quad (6.31)$$
$$s.t. \quad z = Am - b$$

In the following proposition, we show that using the linearized loss $\tilde{l}_t(m) = m^\top g_t$, $\psi_t(m) = \frac{1}{2}(\|m\|_2^2 - \rho\eta_t t\|A(m - \overline{m}_t)\|_2^2)$, and $\mathcal{L}_{\rho,t}$ as the augmented Lagrangian of Problem 6.31, the RDA-ADMM procedure given by Equations (6.28-6.30) admits closed-form solutions.

Proposition 6.1. *Equations (6.28-6.30) where* $\mathcal{L}_{t,\rho}$ *is the augmented Lagrangian of Problem 6.31 with* $\tilde{l}_t(m) = m^\top g_t$*,* $\psi_t(m) = \frac{1}{2}(\|m\|_2^2 - \rho\eta_t t\|A(m - \overline{m}_t)\|_2^2)$*,* $\mathcal{M} = \mathbb{R}^d, \mathcal{Z} =$

$\mathbb{R}^p$, admit the following closed-form solutions:

$$m_{t+1} = -\eta_t t \Big[|\bar{g}_t^\mu| - \lambda\Big]_+ * \operatorname{sign}(\bar{g}_t^\mu) \tag{6.32}$$

$$z_{t+1} = -\Big[\frac{\mu_t}{\rho} - (Am_{t+1} - b)\Big]_+ \tag{6.33}$$

$$\mu_{t+1} = \mu_t - \rho(Am_{t+1} - z_{t+1} - b) \tag{6.34}$$

with $\bar{g}_t^\mu = \bar{g}_t - A^\top(\bar{\mu}_t - \rho(A\bar{m}_t - \bar{z}_t - b))$.

Proof. First we give a simplified expression of $\mathcal{L}_{t,\rho}$:

$$\begin{aligned}\mathcal{L}_{t,\rho}(m,z,\mu) &= \bar{g}_t^\top m + \lambda\|m\|_1 + \frac{1}{2\eta_t t}\|m\|_2^2 + \mathbb{I}_-(z) - \mu^\top(Am - z - b) + \frac{\rho}{2}\|Am - z - b\|_2^2 \\ &- \frac{\rho}{2}\|A(m - \bar{m}_t)\|_2^2 \\ &= \bar{g}_t^\top m + \lambda\|m\|_1 + \frac{1}{2\eta_t t}\|m\|_2^2 + \mathbb{I}_-(z) - \mu^\top(Am - z - b) + \rho(A\bar{m}_t - z - b)^\top Am \\ &+ \frac{\rho}{2}(\|z + b\|_2^2 - \|A\bar{m}_t\|_2^2)\end{aligned}$$

Therefore, we obtain the following simplified update of m_t (Equation 6.28) by deleting the terms in $\mathcal{L}_{t,\rho}$ that do not depend on variable m:

$$\begin{aligned}m_{t+1} &= \arg\min_m \; \mathcal{L}_{t,\rho}(m, \bar{z}_t, \bar{\mu}_t) &(6.35)\\ &= \arg\min_m \; \bar{g}_t^\top m + \lambda\|m\|_1 + \frac{1}{2\eta_t t}\|m\|_2^2 - (\bar{\mu}_t - \rho(A\bar{m}_t - \bar{z}_t - b))^\top Am \\ &= \arg\min_m \; \bar{g}_t^{\mu\top} m + \lambda\|m\|_1 + \frac{1}{2\eta_t t}\|m\|_2^2 \\ &= \arg\min_m \; \frac{1}{2}\|m + \bar{g}_t^\mu \eta_t t\|_2^2 + \lambda\eta_t t\|m\|_1 \\ &= \operatorname{prox}_{\lambda\eta_t t\|\cdot\|_1}(-\bar{g}_t^\mu \eta_t t) &(6.36)\end{aligned}$$

with $\bar{g}_t^\mu = \bar{g}_t - A^\top(\bar{\mu}_t - \rho(A\bar{m}_t - \bar{z}_t - b))$. Therefore, using Equation 6.8, we obtain:

$$m_{t+1} = -\eta_t t \operatorname{sign}(\bar{g}_t^\mu)\Big[|\bar{g}_t^\mu| - \lambda\Big]_+$$

Now, we give the closed-form solution of the update of z_t (Equation 6.29). By *deleting the terms that do not depend on variable z in $\mathcal{L}_{t,\rho}$, Equation 6.29 boils down to:*

$$\begin{aligned}
z_{t+1} &= \arg\min_z \; \mathcal{L}_{t,\rho}(m_{t+1}, z, \mu_t) \\
&= \arg\min_z \; z^T \mu_t + \frac{\rho}{2}\|Am_{t+1} - z - b\|_2^2 + \mathbb{I}_-(z) \\
&= \arg\min_z \; \frac{\rho}{2}\|Am_{t+1} - z - b - \frac{\mu_t}{\rho}\|_2^2 + \mathbb{I}_-(z) \\
&= \operatorname{prox}_{\mathbb{I}_-}(Am_{t+1} - b - \frac{\mu_t}{\rho}) \\
&= -\left[\frac{\mu_t}{\rho} - (Am_{t+1} - b)\right]_+
\end{aligned}$$

where we used Lemma 3 of Appendix C in the last line. Finally, the update of μ_t (Equation 6.30) is already given analytically.

It is important to note that the inclusion of a term $\|A(m - \overline{m}_t)\|_2^2$ in ψ_t, as in Theorem 6.2, is a standard linearization trick in ADMM methods [Deng and Yin, 2016, Wang and Banerjee, 2012, Suzuki, 2013]. It indeed allows bypassing the non-separability of the optimization problem given by equation (6.28), induced by the term $\frac{\rho}{2}\|Am - z - b\|_2^2$. Such a linearization is sometimes referred to as *the split inexact Uzawa method* [He and Yuan, 2012, Zhang et al., 2011].

An online sparse and constrained preference learning algorithm Then, by taking $A = C$ where C is the matrix encoding monotonicity/supermodularity constraints as in Example 6.2, $b = 0$ and g_t as the subgradients of the pref hinge loss (see Equation 6.18), exploiting Proposition 6.1, we obtain an online algorithm for learning sparse and constrained Möbius vector in model F_m. The algorithm is explicitly given in Algorithm 6.3 for $\eta_t = \frac{\gamma}{\sqrt{t}}$, where Equations 6.32-6.34 corresponds to line 9-11.

The benefit of Algorithm 6.3 for retrieving monotonic/supermodular capacity is illustrated in the next section with numerical experiments. While the algorithm performs well in practice, a regret analysis remains to be established. This question is addressed in a preliminary manner in the following paragraph.

Regret and constraint violation bound analysis The update given by Equation 6.31 does not fall within the scope of Theorem 6.2 for several reasons.

(a) assumption (ii) is incompatible with the efficient update formulas provided in Proposition 6.1, which are valid for $\mathcal{M} = \mathbb{R}^d$ and $\mathcal{Z} = \mathbb{R}^p$. A possible option is to derive a regret analysis using a lighter assumption—for instance, solely bounding the distance between the initial model and the fixed model, as in [Wang and Banerjee, 2012] (i.e., $\|m_1 - m\|_2 \leq D$ and $\|z_1 - z\|_2 \leq D$).

Algorithm 6.3: RDA-ADMM for constrained capacity learning

```
Inputs: $(\gamma, \lambda, \rho, T)$
$t \leftarrow 1, \quad m_1, \mu_1, z_1 \leftarrow (0, \dots, 0)$
while $t < T$ do
    receive pairwise example $(x_t, x'_t)$
    compute loss gradient $g_t \in \partial l_t(m_t)$ according to Equation 6.18
    # update average gradient
    $\bar{g}_t \leftarrow \frac{t-1}{t} g_{t-1} + \frac{1}{t} g_t$
    $\bar{g}_t^{\mu} \leftarrow \bar{g}_t - C^{\top}(\bar{\mu}_t - \rho(C\bar{m}_t - \bar{z}_t))$
    # update model
    $m_{t+1} \leftarrow -\gamma\sqrt{t}\Big[|\bar{g}_t^{\mu}| - \lambda\Big]_+ * \operatorname{sign}(\bar{g}_t^{\mu})$
    $z_{t+1} \leftarrow -\Big[\frac{\mu_t}{\rho} - Cm_{t+1}\Big]_+$
    $\mu_{t+1} \leftarrow \mu_t - \rho(Cm_{t+1} - z_{t+1})$
    $t \leftarrow t + 1$
Outputs: $m_T$
```

(b) assumption (*iii*) is too restrictive as it does not allow taking $u = \mathbb{I}_{-}$. Indeed, since for any $z \in$ *dom* u (i.e., $z \leq 0$), it can easily be checked from the subgradient definition (see Definition 1.27) that:

$$\partial u(z) = \{(s_1, \dots, s_p) \in \mathbb{R}_+^p : s_i = 0 \text{ whenever } z_i < 0\}$$

$\partial u(z)$ corresponds to the *normal cone* of the negative orthant at point z, which is represented for $p = 2$ in Figure 6.1 for the sake of illustration. Therefore, $\partial u(z)$ is unbounded, preventing compliance with assumption (*iii*). The fact that neither the batch ADMM analysis [He and Yuan, 2012, Wang and Banerjee, 2012] nor the online ADMM analysis for the OMD procedure [Wang and Banerjee, 2012] necessitates such an assumption suggests that it could potentially be omitted.

(c) as the motivation for Theorem 6.2 is the setting (1) of RDA-ADMM (see Section 2.2.3), it is thought of for a ℓ_1-regularization carried by the u function, and the case of a composite ℓ_1-regularized loss l_t as in Problem 6.31 is not considered. Therefore, the previous points set aside, a direct application of Theorem 6.2 would yield a regret bound in $(G + \lambda d)\sqrt{T}$, as explained in Remark 6.2. Then, an adapted analysis is required, as provided in Xiao [2010] (Corollary 2) or Orabona [2019] (Section 7.8)) for FTRL with composite loss.

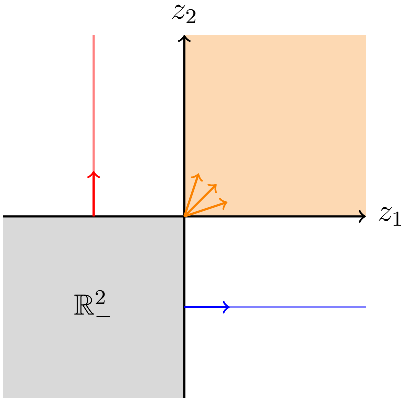


Figure 6.1: $\mathbb{R}^2_-$ normal cone at $(-1, 0)$ (red), $(0, -1)$ (blue) and $(0, 0)$ (orange)

3 Numerical Tests

In this section, we conduct numerical tests using synthetic preference data We generate preference data by randomly drawing sparse (with few non-null coefficients) normalized Möbius vector m associated with monotonic capacities and pairs of alternatives $x_t, x'_t \in [0, 1]^n$. Then, after comparison of the perturbed overall values $m^\top \phi(x_t) + \epsilon_x$ and $m^\top \phi(x'_t) + \epsilon_y$ (where ϵ_x is a centered Gaussian noise with standard error $\sigma = 0.03$), we obtain preference or indifference examples. In all the experiments, we test our algorithms on the learning of Choquet Integral, and thus we generate data using $\phi(x_t) = (\min_{i\in S}\{x_i\})_{S\subseteq N}$ but the tests could be presented with the multilinear model with similar results.

In the first experiment, we show the practical efficiency of Algorithm 6.2 compared to batch problem (6.19) solved with linear programming (denoted Batch(LP)). The ℓ_1-regularization parameter λ is set to 0.01 for both methods and for Algorithm 6.2, γ is set to 10^{-3}. In Table 6.2 and 6.3 we compare the average accuracy and training times over 20 simulations of both methods for a growing number of criteria n. The accuracy is computed as the average proportion of correctly predicted preferences within a test set containing 500 preference examples. The number of preference examples T increases linearly with n. We observe that for 10 and 15 criteria, Algorithm 6.2 reaches accuracy values close to the one obtained with the batch solution (at most 5% lower) while having significantly lower training times. Finally, for 20 criteria (millions of possible criteria interactions), it provides a solution in around 1 minute that approximately reaches 80% of accuracy while no solution can be obtained in batch using linear programming.

In the second experiment, we first compare Algorithm 6.2 and Algorithm 6.3 in the retrieval of monotonic capacities. The number of criteria is set to $n = 10$ and the total number of preference examples is set to $T = 1000$. Preference examples

(n, T)	(10, 500)	(15, 750)	(20, 1000)
Batch (LP)	0.92 ± 0.03	0.89 ± 0.03	–
Algo 1	0.88 ± 0.03	0.84 ± 0.03	0.79 ± 0.03

Table 6.2: Average accuracy over 20 simulations.

(n, T)	(10, 500)	(15, 750)	(20, 1000)
Batch (LP)	1.94 ± 0.21	246.8 ± 20.4	–
Algo 1	0.04 ± 0.01	0.7 ± 0.1	66.7 ± 1.9

Table 6.3: Average training times (sec.) over 20 simulations.

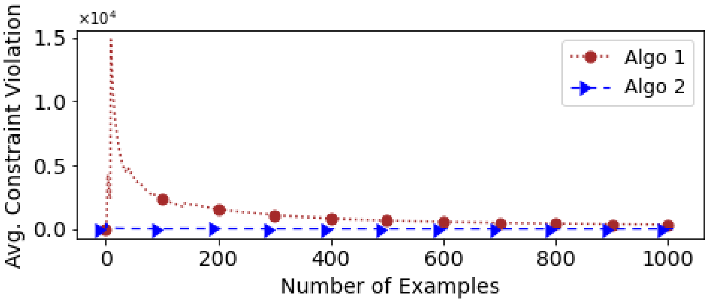


Figure 6.2: Average constraint violation w.r.t. the number of preference examples t.

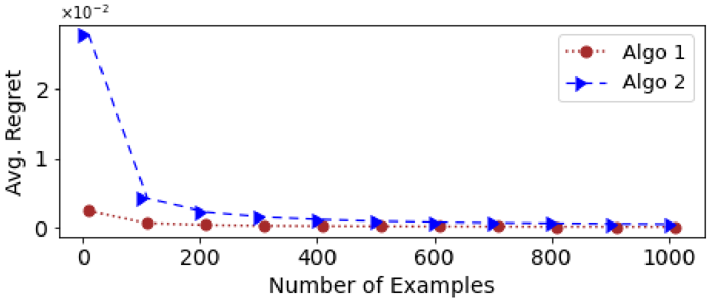


Figure 6.3: Average regret w.r.t. the number of preference examples t.

are generated as in the previous experiment; hyper-parameters λ and γ are unchanged and $\rho = 1$ for Algorithm 6.3. Figure 6.2 represents the average monotonicity violation computed as $\frac{1}{t}\sum_{\tau=1}^{t}\|[Cm_\tau]_+\|_2^2$ where C is the matrix encoding the monotonicity constraints. We observe that Algorithm 6.2 highly violates monotonicity constraints before $t = 200$ examples while we obtain a nearly null average violation for Algorithm 6.3 at any t. Remark that Algorithm 6.2 progressively captures monotonicity as it receives preference examples. In Figure 6.3, we show the average regret $\frac{1}{t}R_t = \frac{1}{t}\sum_{\tau=1}^{t}(l_\tau(m_\tau) + \lambda\|m_\tau\|_1) - \min_{m\in\mathcal{M}} \frac{1}{t}\sum_{\tau=1}^{t}(l_\tau(m) + \lambda\|m\|_1)$ w.r.t. the number of pref-

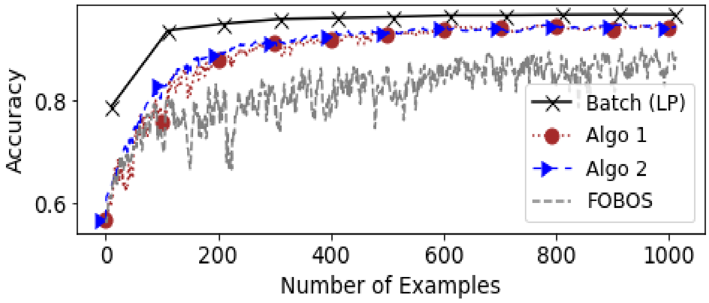


Figure 6.4: Accuracy w.r.t. the number of preference examples t.

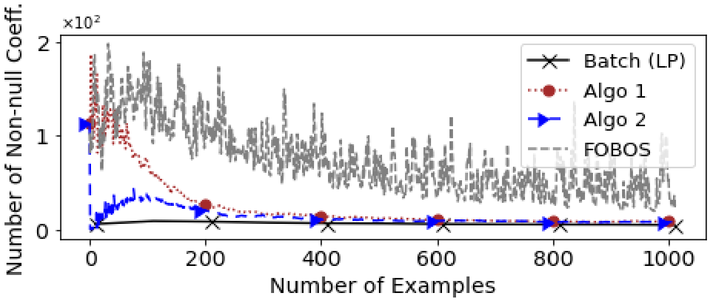


Figure 6.5: Number of non-null coefficients w.r.t. the number of preference examples t.

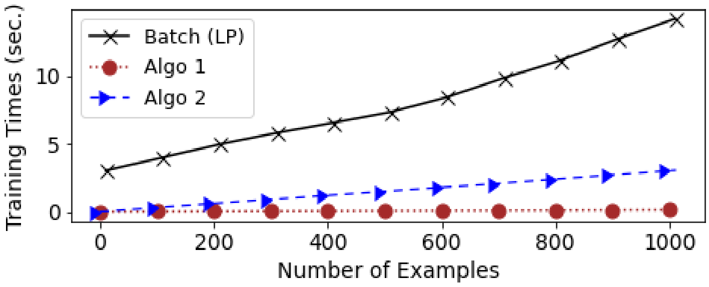


Figure 6.6: Training times (sec.) w.r.t. the number of preference examples t.

erence examples t. The optimal value $\min_{m \in \mathcal{M}} \frac{1}{t} \sum_{\tau=1}^{t} (l_\tau(m) + \lambda \|m\|_1)$ is computed with linear programming. We observe that both algorithms provide sequences of learned models m_t with vanishing average regrets.

Then, we show the performances of both Algorithms 6.2 and 6.3 in terms of accuracy and number of non-null coefficients respectively in Figure 6.4 and 6.5. We compare their performances with the Batch(LP) method and with the FOBOS algorithm implemented using the loss l_t and a learning rate $\eta_t = \gamma / \sqrt{t}$ with γ set at the recommended value in [Duchi and Singer, 2009]. We observe that FOBOS suffers from instability and produces less compact models. In contrast, Algorithms 6.2 and 6.3 quickly reduce the number of non-null coefficients to some dozens. Concerning accuracy, we observe that Algorithm 6.3 achieves the same performance as Algorithm 6.2 while providing a better control of motonicity. The accuracy of both Algorithms 6.2 and 6.3 are slightly below the one obtained with Batch(LP). However, the associated training time curves presented in Figure 6.6 reveal the efficiency of the online algorithms compared to batch (LP). In particular Algorithm 6.2 achieves these results in a near null training time. Algorithm 6.3 achieves intermediate times between Algorithm 6.2 and Batch(LP).

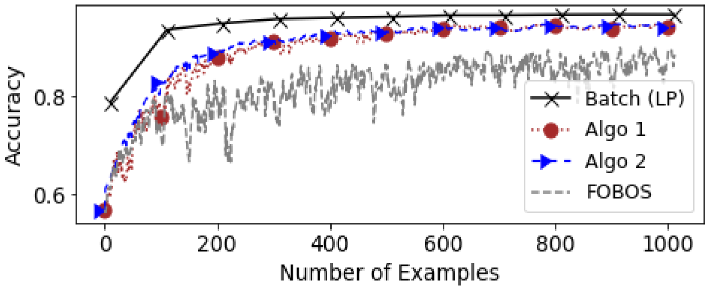


Figure 6.7: Average constraint violation w.r.t. the number of preference examples t.

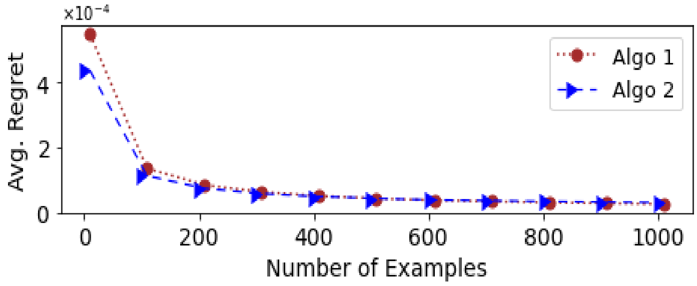


Figure 6.8: Average regret w.r.t. the number of preference examples t.

In the third experiment, we assess the benefit of using Algorithm 6.3 to learn both monotonic and supermodular capacities. More precisely, we compare the average violation of constraints for both Algorithms 6.2 and 6.3 in Figure 6.7 and the average regret in Figure 6.8. The advantage of Algorithm 6.3 in terms of constraint satisfaction is also clear when supermodularity is required in addition to monotonicity.

4 Conclusion

We have proposed online algorithms to efficiently learn the capacity in a large class of non-linear aggregation functions (but linear in the capacity), including the well-known Choquet and multilinear models. These algorithms not only allow a decision model to be adapted to a stream of preference examples, but can also be used in place of batch learning methods, with an advantage in terms of scalability confirmed by our tests. We have also addressed the inclusion of normative constraints restricting the set of admissible capacities in the online learning process.

A direct follow-up to this work would be to derive a regret analysis for Algorithm 6.3. On the experimental side, one could assess the practical ability of the proposed algorithms to adapt to time-varying preferences. Furthermore, a promising long-term direction for this work would be to investigate the potential benefit of active selection of the next example in this online process, while maintaining the computational efficiency of the model update at each iteration, thus achieving a *computationally efficient active learning* [Awasthi et al., 2015, Zhang, 2018, Zhang et al., 2020]. Also, further contributions could involve finding equivalents of the proposed approach for models beyond the class represented by the F_m model. Other aggregation functions based on different algebraic operations can indeed be used to combine capacities and values. For example, *Sugeno's integral* (see Definition 1.17) uses $(\max, \min)$ operations instead of $(+, \times)$ [Sugeno, 1977]. The main challenge in going beyond F_m will then be to overcome the loss of linearity of the model with respect to the capacity and its Möbius inverse m.

Conclusion

Synthesis

The objective of this thesis was to provide learning methods to obtain representations of preferences that are:

- *structured* through to the use of axiomatically grounded models from *decision theory*, which ensure a certain degree of consistency and rationality in preferences,
- and *expressive* enough to capture complex and diverse behavior, through algorithmic tools from *machine learning* and *optimization* enabling the full but controlled exploitation of the descriptive richness of the decision-theoretic models.

To this end, we first introduced in *Chapter 1* the preference models studied in this thesis, i.e., utility functions that allow for interactions between viewpoints, as well as the fundamentals of supervised learning and the optimization techniques used to solve the learning problems. We then presented several contributions related to the learning of multiple utility models in various contexts: from pre-collected datasets of examples (passive learning), from the answers to carefully selected queries (preference elicitation or active learning), or from streams of preference examples (online learning). The contributions are summarized below according to the utility model considered.

The Choquet integral of marginal utilities (CIU) In *Chapter 2*, we addressed the learning of CIU, a central model in decision theory, which involves an initial challenge: disentangling the marginal utility functions from the capacity weights, when neither is directly observable. Hence, we proposed a standard-sequence-based method that uses carefully selected queries to extract information on marginal utilities taking the form of a set of linear constraints on the latter, which are then used to fit spline functions. More robust to response errors than the classical standard-sequence methods initially proposed for eliciting the RDU [Quiggin, 2012] or CPT [Kahneman and Tversky, 1979] model, they offer the additional benefit of being valid for the bipolar Choquet integral (including CIU). The second challenge concerns the identification of the capacity, defined by an exponential number of weights (in n the number of viewpoints), and thus whose flexibility must be properly controlled to ensure that the model fits the data well, while generalizing well to new examples and remaining easy to interpret. Rather than using standard approaches that restrict the flexibility of the capacity a priori (e.g., using k-additivity or predefined hierarchical structures) and may drastically limit the expressiveness of CIU, we proposed learning a sparse representation of the capacity using sparsity-inducing regularizations. More precisely, our approach employs the Möbius transform of the capacity, chosen for its ability to yield sparse capacity representations, and identify the Möbius transform minimizing both the error on the examples and the ℓ_1-norm (possibly weighted). Experiments on synthetic and real-world datasets show that, by tuning the regularization parameter, our method achieves better trade-offs between model simplicity and generalization performance compared to approaches based on structural restrictions like k-additivity.

Capacity-based preference models The learning problem considered in Chapter 2, which aims to find sparse Möbius representations of the capacity can be solved with high precision using linear programming but becomes intractable when n exceeds a dozen due to the exponential growth in n of the number of variables. Thus, *Chapter 3* introduces a method based on the IRLS (iteratively reweighted least squares) paradigm, suitable for learning a large class of capacity-based preference models including CI and the multilinear utility. More precisely, the proposed approach reformulates the problem into a sequence of least squares problems, which admit a compact dual formulation akin to that of support vector machines — a quadratic program whose number of parameters and constraints is linear in the number of examples and independent of n. Experiments on synthetic data show that it allows approximately solving the learning problem for n exceeding 20, i.e., for over a million possible interactions, and hundreds of preference examples. Moreover, experiments on real-world data, conducted in collaboration with domain experts, demonstrated that the algorithm is able to uncover meaningful interactions.

In *Chapter 6*, the learning of capacity-based preference models is addressed in the online setting, where preference examples are revealed sequentially. In particular, we adapt the RDA (regularized dual averaging) online learning algorithm to the learning of sparse Möbius representations, which consists in identifying, upon the arrival of each new example, the vector that minimizes the cumulative error on the examples observed so far, along with an ℓ_1 regularization term. Using a linearization of the error term that preserves the algorithm's regret guarantees, the problem solved at each iteration can be shown to admit a closed-form solution. As our experiments show, this allows for learning capacities with $n = 20$ and 1,000 examples in about one second. Additionally, building on ADMM (alternating direction method of multipliers), we proposed a variant of RDA that incorporates long-term enforcement of constraints on capacity such as monotonicity or supermodularity. Experimental results show that this method allows avoiding large constraint violations along the learning process.

Decision-focused learning with general aggregation functions In *Chapter 5*, we addressed the multi-criteria choice problem, in interaction with the DM. For this problem, a popular strategy is the min-max regret approach that involves querying the DM and progressively narrowing the set of admissible parameter values based on her answers, until an alternative emerges as necessarily optimal. Although efficient in minimizing the number of queries, this type of approach is inherently intolerant to errors in the DM's answers and may not provide a model that is a good representation of her preferences. Hence, we proposed a hybrid algorithm that uses a disagreement-based active learning principle to more safely narrow down the set of admissible parameter values (limiting the possibility of excluding the value that best represents the DM's preferences), while also constructing a dataset from the DM's answers to further identify this optimal parameter value. Since the theoretical guarantees of the algorithm require the use of the 0-1 loss whose minimization is intractable, it is implemented using a discrete set of admissible parameter values. This opens the door to models that have not yet been considered in this thesis, particularly aggregation functions that are nonlinear in their parameters. Numerical experiments show that the proposed approach provides a significant gain in robustness to noisy answers in the identification of the DMs optimal alternative, both for aggregation functions that are linear in their parameters (such as CI) and nonlinear ones, such as the Chebyshev norm.

GAI-decomposable utility functions In *Chapter 4*, we go beyond the framework of totally decomposable models and consider GAI-decomposable utility functions. The expressiveness this utility model —any utility function admits a GAI-decomposition— makes it powerful, but also challenging as the absence of a unique GAI decomposition renders the learning process ill-posed. Therefore, we first proposed to consider standard functional decompositions such as the classical or the anchored ANOVA decompositions to remove any ambiguity in the decomposition's identification problem. Then, the learning of the selected decomposition is performed by using kernel methods. In particular, we use a mutiple kernel learning formulation, that allows learning a sparse decomposition (using as few factors as possible), by solving a quadratically constrained convex optimization problem. This allows learning simultaneously the decomposition and the utility functions defined over the factors, without any assumption on the degree of interactions and for both continuous and discrete attributes, something that was not achieved in the literature on GAI-decomposable utility functions for now.

Future Research Directions

The following section discusses potential extensions of the work presented in this thesis.

Exploring other preference models

- An interesting avenue of research concerns the learning of the Sugeno integral (SI) [Sugeno, 1977], which is often regarded as the ordinal counterpart of the Choquet integral (CI) and is also based on a capacity. Indeed, solving an empirical risk minimization problem formulated with SI poses a significant challenge from an optimization perspective, since, unlike CI or the multilinear utility (MU), it is not linear in the capacity that is involved in min and max operations. More specifically, such optimization problems may be non-convex, and identifying a global minimum requires the development of appropriate optimization methods [Gagolewski et al., 2019b]. Existing approaches are scarce [Prade et al., 2009, Beliakov and Divakov, 2020, Abbaszadeh and Hüllermeier, 2020, Baaj, 2024] and often rely on k-maxitivity [Grabisch, 1997a], the qualitative counterpart of k-additivity, to control the flexibility of the capacity prior to learning. It would therefore be highly relevant to develop methods that adapt the flexibility of the capacity to the data and yield compact representations that do not rely on such restrictions.

- The methods proposed in this thesis could be directly applied to the learning of *compare-and-aggregate* preference models, which have the advantage of being able to describe non-transitive preferences. For instance, we could consider the model $x \succsim x' \Leftrightarrow C_w(x - x') \geq 0$ for any $x, x' \in \mathcal{X}$, where C_w denotes the CI associated with capacity w. Alternatively, we could consider *capacity-based concordance rules* [Dubois et al., 2003], i.e., $x \succsim x' \Leftrightarrow w(c(x, x')) \geq w(c(x', x))$ where $c(x, x')$ denotes the subset of criteria w.r.t. which x is at least as good as x' for any $x, x' \in \mathcal{X}$.

- In the case of regression tasks involving attributes that are not necessarily ordered, or ordered attributes on which the restricted preference $\succsim_i$ is not monotonic with respect to the natural order defined on the attribute, requiring the consistency of the learned model with Pareto dominance is no longer relevant, and we can therefore free ourselves from monotonicity constraints on the capacity. Clearly, dropping monotonicity constraints should provide a significant computational gain, but the associated descriptive gain is unclear. It would be interesting to study to what extent regressions without monotonicity constraints via an integral using a non-monotonic capacity improve the descriptive power of monotonic models.

Limits of ℓ_1 regularization in terms of variable selection In this thesis, we have demonstrated the practical advantage of using the ℓ_1 regularization to learn sparse representations of the capacity in models such as CI or MU. That said, we also observed how the statistical limitations of this regularization can negatively impact the quality of the learned representations, especially when it comes to variable selection. More precisely, as discussed in Chapter 2, when learning a sparse Möbius transform, correlations between the components $\Phi_S = \min_{i \in S}\{x_i\}, S \subseteq N$ can undermine the ability of an ℓ_1-regularized regression to accurately recover the true set of non-zero Möbius masses from a hidden model. Although numerical tests suggest that adaptive ℓ_1 regularization can compensate for this weakness in practice, it does require computing the weights involved in the regularization in advance and tuning an additional hyperparameter. An interesting research direction would be to compare the latter regularization with ℓ_0 regularization, which penalizes the number of non-zero coefficients. Unlike ℓ_1-based regularizations, the ℓ_0 regularization does not introduce bias by penalizing coefficients with large magnitude more severely (and thus shrinking coefficients toward zero) [Bertsimas et al., 2016]. While minimizing the ℓ_0-norm was long considered intractable, recent advances in mixed integer programming have opened up the possibility of applying such methods to high-dimensional problems [Bertsimas and Parys, 2020, Guyard et al., 2024], suggesting that capacity learning with this type of approach could be possible.

Beyond sparsity: toward general compact representations of capacities The search for sparse Möbius transforms for compactly encoding the exponential number of coefficients defining a capacity was motivated in Chapter 2. Yet, one may consider using a generalized notion of compactness by seeking Möbius transforms with a limited number of distinct coefficient values (not necessarily zero). For example, symmetric capacities (which yield OWA operators within the Choquet integral) are described in their Möbius transform with only n distinct values (corresponding to subset sizes), despite the fact that they are not sparse if these values are non-zero. A way to induce this type of structure could be to use a *fused* LASSO regularization [Tibshirani et al., 2005, Sokolovska et al., 2017] of the form $\lambda \sum_{j=1}^{p-1} |\beta_{j+1} - \beta_j|$, which encourages equality between adjacent coefficients (potentially lexicographically ordered here).

Exploiting label complexity to bound the number of queries in incremental preference elicitation In Chapter 5, we adopted a disagreement-based active learning strategy for the incremental elicitation of preferences in the context of a multi-criteria choice problem. Nevertheless, we did not take advantage of *label complexity* results, which are fundamental in the active learning theory [Hanneke et al., 2014]. Such results provide upper bounds on the number of labeled examples needed so that, with high probability, the learned model's true risk is within an ϵ margin of the optimal model. Leveraging such bounds to derive guarantees on the number of queries needed to identify the DM's optimal alternative with a given precision would be an interesting direction, especially since no such result currently exists for incremental elicitation methods based on minimax regret strategy (even though, in practice, the number of queries remains relatively low [Benabbou et al., 2017b, Bourdache et al., 2019b]). It is important to note that sample complexity results, for instance those associated with disagreement-based algorithms [Hanneke et al., 2014], depend on the expressiveness of the model class, often characterized by its VC-dimension [Vapnik, 1995]. This would thus require a thorough analysis, and possibly an extension of existing results on the VC-dimension of the utility function models in decision theory [Hüllermeier and Fallah Tehrani, 2012, Basu and Echenique, 2020].

Preference learning for several DM In contexts involving multiple DM (e.g., music applications, movie streaming platforms), a naive adaptation of the learning algorithms proposed in this thesis would consist in running the algorithms separately for each DM (or user), using their own dataset of preference examples. However, the amount of data available for each user is likely to be small, which may result in poor performance of the learned models. Yet, users could benefit from leveraging other users' data, as common patterns may exist beyond individual preferences (e.g., popular songs are liked by everybody). One approach to address this issue while preserving the privacy of each user's data is to use *federated learning* [McMahan et al., 2017, Kairouz et al., 2021], which enables collaborative model training without data sharing. This can be achieved by performing local model updates on each user's device using their personal data, and sending only the model parameters to a central server. The server then aggregates the parameters from all users and sends back a global model to each user. Since such an approach is designed to learn a single global model, it may not be well-suited to capturing individual user behaviors. A more appropriate setting could be that of *personalized* federated learning. For example, one possible approach is to use the *model-agnostic meta-learning* framework [Fallah et al., 2020], which seeks a shared initialization that captures common patterns across users, such that each user's can quickly adapt the model to their local data with just a few gradient updates.

A different but related research direction could involve evaluating the benefits of using a general aggregation function, such as the Choquet or Sugeno integral, to combine model parameters at the server level. Some initial experiments have already been conducted [Pękala et al., 2024], showing improved performance compared to the standard weighted average, and suggesting that a more refined aggregation taking into account positive or negative synergies between users might be valuable.

Interpretability methods for machine learning Capacities, or more generally, *games* (i.e., set functions that are not necessarily monotonic), play an important role in interpretability methods in machine learning, particularly in approaches that aim to quantify the relative importance of features for a trained model f. Among the key methods, one can cite *kernelSHAP* [Lundberg and Lee, 2017], which models the importance of groups of features in the prediction of a value $f(x)$ using a game $w(S)$, defined for instance as $w(S) = \mathbb{E}[f(z) \mid x_S]$. This modeling approach allows for assigning an importance score to each feature via the Shapley values of the game. The Shapley values can be computed by estimating the game for some coalitions, and searching for the additive game that best approximates the observations in the sense of a (weighted) least squares optimization problem. To obtain more refined representations of the role of features, some works have attempted to generalize this approach by approximating the observed game with k-additive games [Pelegrina et al., 2023, Fumagalli et al., 2024, Pelegrina et al., 2025]. However, this approach requires fixing the order k of the representation in advance, and results indicate that this choice is crucial. Therefore, an interesting research direction would be to adapt the methods proposed in this thesis to overcome this limitation by seeking a *sparse Möbius* representation, allowing to detect the few interaction components describing the game w, whatever the order.

Appendix A

A.1 Q-queries without the Restricted Solvability Assumption

In this section we consider the case where restricted solvability w.r.t component i does not hold, i.e., when exact answers to queries Q_{ij} do not necessarily exist. In particular, we consider the case of discrete attributes (the most common case where restricted solvability fails to hold). The elements of X_i are denoted $x_{i,k}$ and indexed according to their relative values: $x_{i,k} \precsim_i x_{i,k+1}$, for any k.

(i) Marginal value elicitation below the neutral level

Proposition 7.7. *For any attribute $j \in N$, let r_j, $R_j \in X_j$ and $x_i \in X_i$ such that $\mathbf{0}_j \precsim_j r_j \prec_j R_j$, and $x_i \precsim_i \mathbf{0}_i$. If the two following queries are successively asked:*

- *what is the lowest k such that $(x_i, r_j, \mathbf{0}_{-ij}) \precsim (x_{i,k+1}, R_j, \mathbf{0}_{-ij})$? Then we set $y_i^+ = x_{i,k+1}$ and $y_i^- = x_{i,k}$.*
- *what is the highest k such that $(y_i^+, r_j, \mathbf{0}_{-ij}) \succsim (x_{i,k}, R_j, \mathbf{0}_{-ij})$? Then we set $z_i^- = x_{i,k}$ and $z_i^+ = x_{i,k+1}$.*

then, the following inequalities hold:

$$u_i(y_i^+) - u_i(z_i^-) \quad \geq \quad u_i(x_i) - u_i(y_i^+) \tag{7.37}$$

$$u_i(y_i^+) - u_i(z_i^+) \quad < \quad u_i(x_i) - u_i(y_i^-) \tag{7.38}$$

Proof. *By construction y_i^- necessarily verifies the following strict preference: $(x_i, r_j, \mathbf{0}_{-ij}) \succ (y_i^-, R_j, \mathbf{0}_{-ij})$. Hence, with $(x_i, r_j, \mathbf{0}_{-ij}) \precsim (y_i^+, R_j, \mathbf{0}_{-ij})$, we obtain the following inequations: $(u_i(x_i) - u_i(y_i^+))(1 - w'(N \setminus \{i\})) \leq (u_i(R_j) - u_i(r_j))w(\{j\})$ and $(u_i(x_i) - u_i(y_i^-))(1 - w'(N \setminus \{i\})) > (u_i(R_j) - u_i(r_j))w(\{j\})$.*

Similarly, z_i^+ verify $(y_i^+, r_j, \mathbf{0}_{-ij}) \prec (z_i^+, R_j, \mathbf{0}_{-ij})$. Hence, with $(y_i^+, r_j, \mathbf{0}_{-ij}) \succsim (z_i^-, R_j, \mathbf{0}_{-ij})$ we obtain the following inequations: $(u_i(y_i^+) - u_i(z_i^-)(1 - w'(N \setminus \{i\})) \geq (u_i(R_j) - u_i(r_j))w(\{j\})$ and $(u_i(y_i^+) - u_i(z_i^+))(1 - w'(N \setminus \{i\})) < (u_i(R_j) - u_i(r_j))w(\{j\})$. Hence we have $(u_i(x_i) - u_i(y_i^+))(1 - w'(N \setminus \{i\})) \leq (u_i(R_j) - u_i(r_j))w(\{j\}) \leq (u_i(y_i^+) - u_i(z_i^-))(1 - w'(N \setminus \{j\}))$.

Moreover, $(u_i(y_i^+) - u_i(z_i^+)(1 - w'(N \setminus \{i\})) < (u_i(R) - u_i(r))w(\{j\}) < (u_i(x_i) - u_i(y_i^-)(1 - w'(N \setminus \{i\}))$*. Assuming* $(-\mathbf{1}_i, \mathbf{0}_{-i}) \prec \mathbf{0}$*, i.e.,* $w'(N \setminus \{i\}) < 1$*, we obtain:*

$$\begin{aligned} u_i(y_i^+) - u_i(z_i^-) &\geq u_i(x_i) - u_i(y_i^+) \\ u_i(y_i^+) - u_i(z_i^+) &< u_i(x_i) - u_i(y_i^-) \end{aligned}$$

Then we overcome the solvability issue by deriving two inequality constraints on the utility function u_i, instead of a unique equality constraint.

(ii) Marginal value elicitation above the neutral level

Proposition 7.8. *For any attribute* $j \in N$*, let* $r_j, R_j \in X_j$ *and* $x_i \in X_i$ *such that* $r_j \prec_j R_j \precsim_j \mathbf{0}_j$ *and* $x_i \succsim_i \mathbf{0}_i$*. If the two following queries are successively asked:*

- *what is the highest k such that* $(x_i, R_j, \mathbf{0}_{-ij}) \succsim (x_{i,k}, r_j, \mathbf{0}_{-ij})$*? Then we set* $y_i^- = x_{i,k}$ *and* $y_i^+ = x_{i,k+1}$*.*

- *what is the lowest k such that* $(y_i^-, R_j, \mathbf{0}_{-ij}) \precsim (x_{i,k+1}, r_j, \mathbf{0}_{-ij})$*? Then we set* $z_i^- = x_{i,k}$ *and* $z_i^+ = x_{i,k+1}$*.*

then, the following inequalities hold:

$$\begin{aligned} u_i(y_i^-) - u_i(z_i^+) &\leq u_i(x_i) - u_i(y_i^-) && (7.39) \\ u_i(y_i^-) - u_i(z_i^-) &> u_i(x_i) - u_i(y_i^+) && (7.40) \end{aligned}$$

Proof. By construction y_i^+ *necessarily verifies the following strict preference:* $(x_i, R_j, \mathbf{0}_{-ij}) \prec (y_i^+, r_j, \mathbf{0}_{-ij})$*. Hence, with* $(x_i, R_j, \mathbf{0}_{-ij}) \succsim (y_i^-, r_j, \mathbf{0}_{-ij})$*, we obtain the following inequations:* $(u_i(x_i) - u_i(y_i^-))w(\{i\}) \geq (u_i(r_j) - u_i(R_j))(1 - w'(N \setminus \{j\}))$ *and* $(u_i(x_i) - u_i(y_i^+))w(\{i\}) < (u_i(r_j) - u_i(R_j))(1 - w'(N \setminus \{j\}))$*.*

Similarly z_i^- *verify* $(y_i^-, R_j, \mathbf{0}_{-ij}) \succ (z_i^-, r_j, \mathbf{0}_{-ij})$*. Hence, with* $(y_i^-, R_j, \mathbf{0}_{-ij}) \precsim (z_i^+, r_j, \mathbf{0}_{-ij})$*, we obtain the following inequations:* $(u_i(y_i^-) - u_i(z_i^+))w(\{i\}) \leq (u_i(r_j) - u_i(R_j))(1 - w'(N \setminus \{j\}))$ *and* $(u_i(y_i^-) - u_i(z_i^-))w(\{i\}) > (u_i(r_j) - u_i(R_j))(1 - w'(N \setminus \{j\}))$*. Hence we have* $(u_i(y_i^-) - u_i(z_i^+))w(\{i\}) \leq (u_i(R_j) - u_i(r_j))(1 - w'(N \setminus \{j\})) \leq (u_i(x_i) - u_i(y_i^-))w(\{i\})$*. Moreover,* $(u_i(x_i) - u_i(y_i^+))w(\{i\}) < (u_i(R_j) - u_i(r_j))(1 - w'(N \setminus \{j\})) < (u_i(h_i) - u_i(z_i^-))w(\{i\})$*. Assuming* $(\mathbf{1}_i, \mathbf{0}_{-i}) \succ \mathbf{0}$*, i.e.,* $w(\{i\}) > 0$*, we obtain:*

$$\begin{aligned} u_i(y_i^-) - u_i(z_i^+) &\leq u_i(x_i) - u_i(y_i^-) \\ u_i(y_i^-) - u_i(z_i^-) &> u_i(x_i) - u_i(y_i^+) \end{aligned}$$

A.2 Covariance Computations

Proposition 7.9. *Let* $\rho^1_{1,3}, \rho^1_{1,2}, \rho^2_{2,3} \in [-1,1]$ *such that* $1-3(\rho^1_{1,3})^2 \neq 0$, *and let* $\Sigma^{\mathbf{11}}$ *and* $\Sigma^{\mathbf{21}}$ *be the following matrices:*

$$\Sigma^{\mathbf{11}} = \begin{pmatrix} 1 & 0 & 0 & \rho^1_{1,3} \\ 0 & 1 & 0 & \rho^1_{1,3} \\ 0 & 0 & 1 & \rho^1_{1,3} \\ \rho^1_{1,3} & \rho^1_{1,3} & \rho^1_{1,3} & 1 \end{pmatrix}, \quad \Sigma^{\mathbf{21}} = \begin{pmatrix} \rho^1_{1,2} & \rho^1_{1,2} & 0 & \rho^2_{2,3} \\ \rho^1_{1,2} & 0 & \rho^1_{1,2} & \rho^2_{2,3} \\ 0 & \rho^1_{1,2} & \rho^1_{1,2} & \rho^2_{2,3} \end{pmatrix}$$

If $\operatorname{sign}(\beta^*_{A_1}) = (1,1,1,1)^\intercal$, *then the following inequality:*

$$|\Sigma^{\mathbf{21}}(\Sigma^{\mathbf{11}})^{-1}\operatorname{sign}(\beta^*_{A_1})| < \mathbf{1}$$

where the inequality holds component-wise, is equivalent to:

$$|2\rho^1_{1,2}(1-\rho^1_{1,3}) + \rho^2_{2,3}(1-3\rho^1_{1,3})| < |1-3(\rho^1_{1,3})^2|$$

Proof. $\Sigma^{\mathbf{11}}$ *can be rewritten as a block-matrix as follows:*

$$\Sigma^{\mathbf{11}} = \begin{pmatrix} M_1 & M_2^T \\ M_2 & M_3 \end{pmatrix} \text{ with } M_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, M_2 = \begin{pmatrix} \rho^1_{1,3} & \rho^1_{1,3} & \rho^1_{1,3} \end{pmatrix}, M_3 = \begin{pmatrix} 1 \end{pmatrix}$$

The Schur complement of $\Sigma^{\mathbf{11}}$ *is* $S = M_3 - M_2^T M_1^{-1} M_2 = 1-3(\rho^1_{1,3})^2 \neq 0$ *and therefore* $\Sigma^{\mathbf{11}}$ *is a positive definite matrix the inverse of which reads as follows:*

$$\begin{aligned}(\Sigma^{\mathbf{11}})^{-1} &= \begin{pmatrix} M_1^{-1} + M_1^{-1}M_2^T S^{-1} M_2 M_1^{-1} & -M_1^{-1}M_2^T S^{-1} \\ -S^{-1}M_2M_1^{-1} & S^{-1} \end{pmatrix} \\ &= \frac{1}{1-3(\rho^1_{1,3})^2} \begin{pmatrix} 1-2(\rho^1_{1,3})^2 & (\rho^1_{1,3})^2 & (\rho^1_{1,3})^2 & -\rho^1_{1,3} \\ (\rho^1_{1,3})^2 & 1-2(\rho^1_{1,3})^2 & (\rho^1_{1,3})^2 & -\rho^1_{1,3} \\ (\rho^1_{1,3})^2 & (\rho^1_{1,3})^2 & 1-2(\rho^1_{1,3})^2 & -\rho^1_{1,3} \\ -\rho^1_{1,3} & -\rho^1_{1,3} & -\rho^1_{1,3} & 1 \end{pmatrix}\end{aligned}$$

Then we finally obtain:

$$\Sigma^{\mathbf{21}}(\Sigma^{\mathbf{11}})^{-1}\,\mathrm{sign}(\beta^*_{A_1}) = \frac{2\rho^1_{1,2}(1-\rho^1_{1,3}) + \rho^2_{2,3}(1-3\rho^1_{1,3})}{1-3(\rho^1_{1,3})^2}\begin{pmatrix}1\\1\\1\end{pmatrix}$$

Therefore, all components of the vector $\Sigma^{\mathbf{21}}(\Sigma^{\mathbf{11}})^{-1}\,\mathrm{sign}(\beta^*_{A_1})$ *have absolute values strictly lower than 1 if and only if* $|2\rho^1_{1,2}(1-\rho^1_{1,3}) + \rho^2_{2,3}(1-3\rho^1_{1,3})| < |1-3(\rho^1_{1,3})^2|$.

In the following, we use the convention that for any subset $S \subseteq N$, $\int_{\mathcal{I}_S} f(z_1,\ldots,z_n)\,dz_S$ denotes the multiple integral of the function f w.r.t. the arguments $z_i, i \in S$ on the hypercube $\mathcal{I}_S = [0,1]^s$. Moreover, for any subset $S \subseteq N$, its cardinal $|S|$ is denoted by s. Finally, we use the fact that if $(Z_1,\ldots,Z_n)$ are n independent random variables, each following a uniform distribution over $[0,1]^n$, then for any subset $S \subseteq N$, the random variable $\Phi_S = \min_{i\in S} Z_i$ follows a Beta distribution with parameters $(1,s)$. Consequently, for any $k \in \mathbb{N}$, we have (see, for instance, [Arnold et al., 2008, Chapter 2]):

$$\mathbb{E}[\Phi_S^k] = \int_{\mathcal{I}_S} \min_{i\in S}\{z_i\}^k\,dz_S = \frac{k!s!}{(k+s)!} \tag{7.41}$$

Lemma 1. *Let* $n \le 3$ *and* $B_1, B_2 \subseteq N$ *such that* $B_1 \cap B_2 = \emptyset$ *and* $B_2 \neq \emptyset$. *For any vector* $(z_j)_{j\in B_2}$ *taking values in* $[0,1]$, *the following equality holds:*

$$\int_{\mathcal{I}_{B_1}} \min_{i\in B_2\cup B_1}\{z_i\}\,dz_{B_1} = \wedge_{B_2} - \frac{b_1\wedge^2_{B_2}}{2} + \frac{b_1(b_1-1)^+\wedge^3_{B_2}}{6} \tag{7.42}$$

with $b_1 = |B_1|$, $\wedge_{B_2} = \min_{i\in B_2}\{z_i\}$ *and* $x^+ = \max\{0,x\}$ *for any* $x \in \mathbb{R}$.

Proof. Firstly, for any $u \in [0,1]$ *and any* $k \in \mathbb{N}$, *we have:*

$$\begin{aligned}\int_0^1 \min\{x,u\}^k\,dx &= \int_0^u \min\{x,u\}^k\,dx + \int_u^1 \min\{x,u\}^k\,dx\\ &= \int_0^u x^k\,dx + \int_u^1 u^k\,dx \qquad (7.43)\\ &= \frac{u^{k+1}}{k+1} + (1-u)u^k = u^k - \frac{k}{k+1}u^{k+1} \qquad (7.44)\end{aligned}$$

Remark that for $B_1 = \emptyset$, *the left-hand term of Equation 7.42 boils down to* $\wedge_{B_2}$ *which is equal to the right-hand term for* $b_1 = 0$. *Suppose now that* $B_1 \neq \emptyset$. *Since* $n \le 3$, $B_2 \neq \emptyset$ *and* $B_1 \cap B_2 = \emptyset$, B_1 *is necessarily a singleton or a pair, then we have:* $b_1 \in \{1,2\}$. *Then let* $(\pi_1,\ldots,\pi_{b_1})$ *be any ordering of the elements of* B_1. *Using Equation 7.44 with*

$u = \min_{i\in(B_1\cup B_2)\setminus\{\pi_1\}}\{z_i\}$, $x = z_{\pi_1}$ and $k = 1$, we have:

$$\int_{\mathcal{I}_{B_1}} \min_{i\in B_2\cup B_1}\{z_i\}\ dz_{B_1} = \int_{\mathcal{I}_{B_1\setminus\{\pi_1\}}} \left(\int_0^1 \min\{z_{\pi_1}, \min_{i\in B_2\cup B_1\setminus\{\pi_1\}}\{z_i\}\}\ dz_{\pi_1}\right) dz_{B_1\setminus\{\pi_1\}}$$
$$= \int_{\mathcal{I}_{B_1\setminus\{\pi_1\}}} \left(\wedge_{(B_1\cup B_2)\setminus\{\pi_1\}} - \frac{\wedge^2_{(B_1\cup B_2)\setminus\{\pi_1\}}}{2}\right) dz_{B_1\setminus\{\pi_1\}} \qquad (7.45)$$

Then if $b_1 = 1$, we have $B_1 \setminus \{\pi_1\} = \emptyset$ and $(B_1 \cup B_2) \setminus \{\pi_1\} = B_2$. Therefore, Equation 7.45 directly yields Equation 7.42. Finally, if $b_1 = 2$, we have $B_1 \setminus \{\pi_1\} = \{\pi_2\}$ and $(B_1 \cup B_2) \setminus \{\pi_1\} = B_2 \cup \{\pi_2\}$. Then, using Equation 7.44 for $u = \min_{i\in B_2}\{z_i\}$, $x = z_{\pi_2}$ and $k \in \{1, 2\}$, we obtain:

$$\int_{\mathcal{I}_{B_1}} \min_{i\in B_2\cup B_1}\{z_i\}\ dz_{B_1} = \int_0^1 \left(\min_{i\in B_2\cup\{\pi_2\}}\{z_i\} - \frac{\min_{i\in B_2\cup\{\pi_2\}}\{z_i\}^2}{2}\right) dz_{\pi_2}$$
$$= \wedge_{B_2} - \frac{b_1\wedge^2_{B_2}}{2} + \frac{b_1(b_1-1)^+\wedge^3_{B_2}}{6}$$

Proposition 7.10. *Let $(Z_1, \ldots, Z_n)$ be independent random variables distributed according to a uniform distribution over $[0, 1]^n$ with $n \leq 3$. Then, for any $S_1, S_2 \subseteq N$ such that $|S_1| = s_1$, $|S_2| = s_2$ and $|S_1 \cap S_2| = s_{12}$, the covariance between $\Phi_{S_1} = \min_{i\in S_1}\{Z_i\}$ and $\Phi_{S_2} = \min_{i\in S_1}\{Z_i\}$ is given by:*

$$\mathrm{Cov}(\Phi_{S_1}, \Phi_{S_2}) = \begin{cases} 0 & \textit{if } s_{12} = 0, \\ \sum_{k=1}^3 g_k(s_{12})\gamma_k(s_1, s_2, s_{12}) - \frac{1}{(s_1+1)(s_2+1)} & \textit{otherwise,} \end{cases} \qquad (7.46)$$

with $g_k(s_{12}) = \frac{k!s_{12}!}{(s_{12}+k)!}$, $\gamma_1 = 1$, $\gamma_2(s_1, s_2, s_{12}) = -\frac{1}{2}((s_1 - s_{12})^+ + (s_2 - s_{12})^+)$ and $\gamma_3(s_1, s_2, s_{12}) = \frac{1}{4}((s_1 - s_{12})^+(s_2 - s_{12})^+) + \frac{1}{6}((s_1 - s_{12})^+(s_1 - s_{12} - 1)^+ + (s_2 - s_{12})^+(s_2 - s_{12} - 1)^+)$.

Proof. Let $S_1, S_2 \subseteq N \setminus \emptyset$. If $S_1 \cap S_2 = \emptyset$, since random variables $(Z_1, \ldots, Z_n)$ are independent, so are Φ_{S_1} and Φ_{S_2}, yielding $\mathrm{Cov}(\Phi_{S_1}, \Phi_{S_2}) = 0$. If $S_1 \cap S_2 \neq \emptyset$, we have:

$$\mathbb{E}[\Phi_{S_1}\Phi_{S_2}] = \int_{\mathcal{I}_{S_1\cup S_2}} \min_{i\in S_1}\{z_i\}\min_{i\in S_2}\{z_i\}\ dz_{S_1\cup S_2} = \int_{\mathcal{I}_{S_2}} \min_{i\in S_2}\{z_i\}\left(\int_{\mathcal{I}_{S_1\setminus S_1\cap S_2}} \min_{i\in S_1}\{z_i\}\ dz_{S_1\setminus(S_1\cap S_2)}\right) dz_{S_2}$$

Then, using Lemma 1 sequentially for $B_1 = S_1 \setminus (S_1 \cap S_2)$, $B_2 = S_1 \cap S_2$ and $B_1 =$

$S_2 \setminus (S_1 \cap S_2)$, $B_2 = S_1 \cap S_2$, we obtain:

$$\begin{aligned}
\mathbb{E}[\Phi_{S_1}\Phi_{S_2}] &= \int_{\mathcal{I}_{S_1 \cap S_2}} \Big(\int_{\mathcal{I}_{S_2 \setminus (S_1 \cap S_2)}} \min_{i \in S_2}\{z_i\} \Big(\wedge_{S_1 \cap S_2} - \frac{(s_1 - s_{12})^+ \wedge^2_{S_1 \cap S_2}}{2} \\
&+ \frac{(s_1 - s_{12})^+ (s_1 - s_{12} - 1)^+ \wedge^3_{S_1 \cap S_2}}{6} \Big) \, dz_{S_2 \setminus (S_1 \cap S_2)} \Big) \, dz_{S_1 \cap S_2} \\
&= \int_{\mathcal{I}_{S_1 \cap S_2}} \Big(\wedge_{S_1 \cap S_2} - \frac{(s_1 - s_{12})^+ \wedge^2_{S_1 \cap S_2}}{2} + \frac{(s_1 - s_{12})^+ (s_1 - s_{12} - 1)^+ \wedge^3_{S_1 \cap S_2}}{6} \Big) \\
&\Big(\wedge_{S_1 \cap S_2} - \frac{(s_2 - s_{12})^+ \wedge^2_{S_1 \cap S_2}}{2} + \frac{(s_2 - s_{12})^+ (s_2 - s_{12} - 1)^+ \wedge^3_{S_1 \cap S_2}}{6} \Big) \, dz_{S_1 \cap S_2}
\end{aligned}$$

where $\wedge_{S_1 \cap S_2} = \min_{i \in S_1 \cap S_2}\{z_i\}$ for any vector $(z_i)_{i \in S_1 \cap S_2}$ valued in $[0,1]$. This expression can be simplified remarking that since $n \leq 3$ and $S_1 \cap S_2 \neq \emptyset$, we have that the cross products $(s_2 - s_{12})^+(s_2 - s_{12} - 1)^+(s_1 - s_{12})^+(s_1 - s_{12} - 1)^+$, $(s_2 - s_{12})^+(s_2 - s_{12} - 1)^+(s_1 - s_{12})^+$ and $(s_1 - s_{12})^+(s_1 - s_{12} - 1)^+(s_2 - s_{12})^+$ necessarily equal zero. Finally, using Equation 7.41 for $S = S_1 \cap S_2$ and $k \in \{2,3,4\}$, we obtain that:

$$\begin{aligned}
\mathbb{E}[\Phi_{S_1}\Phi_{S_2}] &= g_2(s_{12}) - g_3(s_{12})\frac{1}{2}((s_1 - s_{12})^+ + (s_2 - s_{12})^+) + g_4(s_{12})(\frac{1}{4}((s_1 - s_{12})^+(s_2 - s_{12})^+) \\
&+ \frac{1}{6}((s_1 - s_{12})^+(s_1 - s_{12} - 1)^+ + (s_2 - s_{12})^+(s_2 - s_{12} - 1)^+)) \qquad (7.47)
\end{aligned}$$

with $g_k(s_{12}) = \frac{k! s_{12}!}{(s_{12}+k)!}$. Finally, Equation 7.41 yields $\mathbb{E}[\Phi_{S_1}]\mathbb{E}[\Phi_{S_2}] = \frac{1}{(s_1+1)(s_2+1)}$, an we obtain Equation 7.46.

Appendix B

Table 8.4 lists the names of the variables, whether they have been multiplied by 1 or -1 (monotonicity denoted by mono.), and their type (categorical or continuous denoted by cat. and cont.). For more information on the construction of variables, see [Jeandidier et al., 2020, Bourreau-Dubois et al., 2022, Jeandidier, 2024] (in French).

Variable	**Mono.**	**Type**
`offered amount (euros)`	1	cont.
`requested amount (euros)`	1	cont.
`age wife (years)`	1	cont.
`age husband (years)`	1	cont.
`health status of the husband`	-1	cont.
`number of dependent children`	1	cat.
`monthly standard of living gap between spouses (euros)`	1	cont.
`indication of a disagreement over child custody`	1	cat.
`disparity in separate assets between spouses`	1	cont.
`the wife took care of the children and the household`	1	cat.
`divorce for husbands fault`	1	cat.
`length of marriage (years)`	1	cont.
`the wife is claiming compensation for damages`	1	cat.
`common assets of the couple (euros)`	1	cont.
`CA in the form of an annuity`	1	cat.
`matrimonial regime unfavorable to the wife`	1	cat.
`male judge`	1	cat.
`medium cities`	1	cat.
`separate property of the husband (euros)`	1	cont.
`disagreement over the alimony for child support`	1	cat.
`the judge temporarily grants the marital home to the wife`	1	cat.
`the judge order the husband to pay damages`	1	cat.
`the wife contributed to the husband's business activities`	1	cat.
`small towns`	1	cat.
`the wife is eligible for` *aide juridictionnelle à taux plein*	-1	cat.

Table 8.4: Variables signification with monotonicity and type.

Appendix C

Lemma 2. *The proximal operator of the ℓ_1-norm is defined for any $x \in \mathbb{R}^d$ by:*

$$\text{prox}_{\lambda\|\cdot\|_1}(x) = \arg\min_{z\in\mathbb{R}^d} \left(\frac{1}{2}\|z-x\|_2^2 + \lambda\|z\|_1\right)$$

which admits the following closed-form solution:

$$\text{prox}_{\lambda\|\cdot\|_1}(x) = \text{sign}(x) * [|x| - \lambda]_+ \tag{9.48}$$

Proof. For any $x \in \mathbb{R}^d$, we have:

$$\text{prox}_{\lambda\|\cdot\|_1}(x) = \arg\min_{z\in\mathbb{R}^d} \left(\sum_{j=1}^{d}\left(\frac{1}{2}(z_j - x_j)^2 + \lambda|z_j|\right)\right) \tag{9.49}$$

Problem 9.49 is separable across coordinates, and thus reduces to solving the following univariate problem for each coordinate j:

$$\min_{z_j\in\mathbb{R}} \left(\frac{1}{2}(z_j - x_j)^2 + \lambda|z_j|\right) \tag{9.50}$$

Therefore, at the optimum z_j satisfies the following necessary condition for optimality (obtained by putting subgradient to zero):

$$(z_j - x_j) + \lambda s_j = 0 \Longleftrightarrow z_j = x_j - \lambda s_j \tag{9.51}$$

where $s_j \in \partial|.|z_j$ and $\partial|.|(z_j)$ is the set of subgradients of the absolute value function at point z_j. $\partial|.|(z_j)$ is detailed below:

$$\partial|.|(z_j) = \begin{cases} \{1\} & \text{if } z_j > 0, \\ \{-1\} & \text{if } z_j < 0, \\ [-1,1] & \text{if } z_j = 0. \end{cases} \tag{9.52}$$

Then, we consider the three following cases:

- *If $x_j < -\lambda < 0$, for any $s \geq -1$ we have: $x_j - \lambda s \leq x_j + \lambda < 0$. Then, with Equation (9.51), necessarily $z_j = x_j - \lambda s_j < 0$ and we obtain $s_j = -1$. Therefore,* *we have:*

$$\begin{aligned} z_j &= x_j + \lambda \\ &= -(|x_j| - \lambda) \end{aligned}$$

where we use $x_j < 0$ in the second line.

- *If $x_j > \lambda > 0$, for any $s \leq 1$ we have: $x_j - \lambda s \geq x_j - \lambda > 0$. Then, with Equation (9.51), necessarily $z_j = x_j - \lambda s_j > 0$ and we obtain $s_j = 1$. Therefore, we have:*

$$\begin{aligned} z_j &= x_j - \lambda \\ &= |x_j| - \lambda \end{aligned}$$

where we use $x_j > 0$ in the second line.

- *If $|x_j| < \lambda$, suppose $z_j > 0$, then $s_j = 1$ and with Equation (9.51), $z_j = x_j - \lambda \leq 0$ which is a contradiction. Similarly, suppose $z_j < 0$, then $s_j = -1$ and with Equation (9.51), $z_j = x_j + \lambda \geq 0$ which is a contradiction also. Then necessarily $z_j = 0$ and $s_j = \frac{x_j}{\lambda} \in [-1, 1]$.*

The three cases can be summarized as follows:

$$\begin{aligned} z_j &= \begin{cases} -(|x_j| - \lambda) & \textit{if } x_j < -\lambda < 0 \\ |x_j| - \lambda & \textit{if } x_j > \lambda > 0 \\ 0 & \textit{if } |x_j| < \lambda \end{cases} \\ &= \operatorname{sign}(x_j)[|x_j| - \lambda]_+ \end{aligned}$$

Lemma 3. *The proximal operator of $\mathbb{I}_-$ is defined for any $x \in \mathbb{R}^d$ by:*

$$\operatorname{prox}_{\mathbb{I}_-}(x) = \arg\min_{z \in \mathbb{R}^d} \left(\frac{1}{2} \|z - x\|_2^2 + \mathbb{I}_-(z) \right)$$

which admits the following closed-form solution:

$$\operatorname{prox}_{\mathbb{I}_-}(x) = -[-x]_+ \tag{9.53}$$

Remark that it corresponds to the Euclidean projection on the negative orthant.

Proof. For any $x \in \mathbb{R}^d$, we have:

$$\operatorname{prox}_{\mathbb{I}_-}(x) = \arg\min_{z \in \mathbb{R}^d} \left(\frac{1}{2} \sum_{j=1}^{d} \Big((z_j - x_j)^2 + \mathbb{I}_-(z_j) \Big) \right) \tag{9.54}$$

Problem 9.54 is separable w.r.t. components z_j. Therefore, z_j is solution of the following univariate optimization problem:

$$z_j = \arg\min_{z_j \leq 0} \frac{1}{2}(z_j - x_j)^2$$

Let $u_j \in \mathbb{R}_+$ denote the Lagrangian multiplier of the sign constraint on z_j. Then the stationary Karush–Kuhn–Tucker (KKT) condition is:

$$z_j - x_j + u_j = 0 \tag{9.55}$$

Also, the KKT primal feasibility condition gives $z_j \leq 0$ and the KKT complementary slackness condition gives $u_j z_j = 0$. Then, we consider the three following cases:

- *If $x_j > 0$, since $z_j \leq 0$, from Equation (9.55) we necessarily have $u_j > 0$. Then from complementary slackness condition $z_j = 0$.*
- *If $x_j < 0$, since $u_j \geq 0$, from Equation (9.55) we necessarily have $z_j < 0$. Then from complementary slackness condition $u_j = 0$. Then with Equation (9.55), $z_j = x_j$.*
- *If $x_j = 0$, suppose $u_j > 0$, then with Equation (9.55), necessarily $z_j < 0$ and complementary slackness condition does not hold. Then $u_j = 0$ and then with Equation (9.55), $z_j = 0$.*

The three cases can be summarized as follows:

$$\begin{aligned} z_j &= \begin{cases} 0 & \textit{if } x_j > 0 \\ x_j & \textit{if } x_j < 0 \\ 0 & \textit{if } x_j = 0 \end{cases} \\ &= -[-x_j]_+ \end{aligned}$$

Appendix D

List of publications

International Journal Publications

1. Herin M., Perny P. and Sokolovska N.(2025). "Learning Additive Decompositions of Multiattribute Utility Functions". In *Theory and Decision.* https://doi.org/10.1007/s11238-025-10068-6.

2. Herin M., Perny P. and Sokolovska N.(2024b). "Learning Preference Representations based on Choquet Integrals for Multicriteria Decision Making". In *Annals of Mathematics and Artificial Intelligence.* https://doi.org/10.1007/s10472-024-09930-0.

International Conferences with Proceedings

1. Herin M., Perny P. and Sokolovska N.(2024e)."Noise-Tolerant Active Preference Learning for Multicriteria Choice Problems". In *Proceedings of the Algorithmic Decision Theory: 8th International Conference* (ADT-24),New Brunswick, NJ, USA, October 14–16, 2024, pp 191 - 206. https://doi.org/10.1007/978-3-031-73903-313.

2. Herin M., Perny P. and Sokolovska N.(2024d). "Online Learning of Capacity-Based Preference Models". In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence* (IJCAI-24), Jeju, South Korea, August 3-9, 2024, pp 7118-7126. https://doi.org/10.24963/ijcai.2024/787. **Distinguished Paper Award.**

3. Herin M., Perny P. and Sokolovska N.(2024a). "Learning GAI-Decomposable Utility Models for Multiattribute Decision Making". In *Proceedings of the 38th AAAI Conference on Artificial Intelligence* (AAAI-24), Vancouver, Canada, February 20-27, 2024, 38(18) 20412-20149. doi.org/10.1609/aaai.v38i18.30024.

4. Herin M., Perny P. and Sokolovska N.(2023). "Learning Preference Models with Sparse Interactions of Criteria". In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence* (IJCAI-23), Macao, China, August 19-25, 2023, pp. 3786–3794.https://doi.org/10.24963/ijcai.2023/421.

5. Herin M., Perny P. and Sokolovska N.(2022a). "Learning Sparse Representations of Preferences within Choquet Expected Utility Theory". In *Proceedings of the 38th*

Conference on Uncertainty in Artificial Intelligence (UAI-22), Eindhoven, Netherlands, August 1-5, 2022, PMLR 180, pp. 800-810. https://proceedings.mlr.press/v180/herin22a.html

International Conferences/Workshops:

1. Tarissan F., Herin M., Perny P., Isabelle S. (2025), "Leveraging the Choquet Integral for Analyzing Court Decisions in Divorce Cases". In *The European Society for Empirical Legal Studies 2025 Conference (ESELS-25)*, Toulouse, France, June 18-20, 2025.

2. Herin M., Perny P. and Sokolovska N.(2022c). "A Dual Approach for Learning Sparse Representations of Choquet Integrals". In *DA2PL From Multiple-Criteria Decision Aid to Preference Learning*, November 2022, Compiègne, France.

3. Herin M., Perny P. and Sokolovska N.(2022b). "Learning Utilities and Sparse Representations of Capacities for Multicriteria Decision Making with the Bipolar Choquet Integral". In *The 13th Multidisciplinary Workshop on Advances in Preference Handling* (in conjunction with IJCAI-22), July 2022, Vienna, Austria.

National Conferences

1. Herin M., Perny P. and Sokolovska N.(2024c). "A Unified Approach to Learn Decision Models with Interactions". In *25ème congrès annuel de la société française de recherche opérationnelle et d'aide à la décision* (ROADEF-24), 4-7 mars, 2024, Amiens, France. **Best Student Paper**.

Talks & Invited Presentations

1. *UQSay Seminar (Uncertainty Quantification)*, "Algorithms for learning capacity-based preference models" (invited talk), March 2025, https://www.uqsay.org

2. *The* 14th *Multidisciplinary Workshop on Advances in Preference Handling (IJCAI-23 Workshop)*, "Learning Compact Preference Representations based on Choquet Integrals." (contributed talk), Aug. 23, Macao, China, https://sites.google.com.

3. *Seminar Discrete Mathematics, Optimization, Decision-making (CES-University Paris I)* "Learning Compact Preference Representations based on Choquet Integrals for Multicriteria Decision Making." (invited talk), Sept. 2023, https://sites.google.com.

Bibliography

Sadegh Abbaszadeh and Eyke Hüllermeier. Machine learning with the sugeno integral: The case of binary classification. *IEEE Transactions on Fuzzy Systems*, 29(12):3723–3733, 2020. (Cited on pages 52 and 228.)

Mohammed Abdellaoui. Parameter-free elicitation of utility and probability weighting functions. *Management Science*, 46(11):1497–1512, 2000. (Cited on pages 28, 59, and 64.)

Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55. US Government printing office, 1968. (Cited on page 163.)

Loïc Adam and Sébastien Destercke. Handling inconsistency in (numerical) preferences using possibility theory. *Information Fusion*, 103:102089, 2024. (Cited on pages 30, 176, and 177.)

Titilope A. Adeyeba, Derek T. Anderson, and Timothy C. Havens. Insights and characterization of l1-norm based sparsity learning of a lexicographically encoded capacity vector for the Choquet integral. In *FUZZ-IEEE*, pages 1 – 7, 2015. (Cited on pages 3, 52, 53, 58, and 106.)

Manish Aggarwal and Ali Fallah Tehrani. Modelling human decision behaviour with preference learning. *INFORMS Journal on Computing*, 31(2):318–334, 2019. (Cited on page 198.)

Julien Ah-Pine, Brice Mayag, and Antoine Rolland. Additive bi-capacity by using mathematical programming. In *Third International Conference on Algorithmic Decision Theory (ADT)*, pages 15–29, 2018. (Cited on pages 31 and 106.)

Shotaro Akaho. A kernel method for canonical correlation analysis. *arXiv preprint cs/0609071*, 2006. (Cited on page 155.)

Alnur Ali and Ryan J Tibshirani. The generalized lasso problem and uniqueness. 2019. (Cited on page 214.)

Nahla Ben Amor, Didier Dubois, Hela Gouider, and Henri Prade. Graphical models for preference representation: An overview. In *International Conference on Scalable Uncertainty Management*, pages 96–111. Springer, 2016. (Cited on page 142.)

Derek T. Anderson, Stanton R. Price, and Timothy C. Havens. Regularization-based learning of the Choquet integral. In *FUZZ-IEEE*, pages 2519 – 2526, 2014. (Cited on pages 3, 52, 53, 58, and 106.)

Silvia Angilella, Salvatore Greco, Fabio Lamantia, and Benedetto Matarazzo. Assessing non-additive utility for multicriteria decision aid. *European Journal of Operational Research*, 158(3):734 – 744, 2004. (Cited on page 28.)

Silvia Angilella, Salvatore Greco, and Benedetto Matarazzo. Non-additive robust ordinal regression: A multiple criteria decision model based on the choquet integral. *European Journal of Operational Research*, 201(1):277–288, 2010. (Cited on page 30.)

Silvia Angilella, Salvatore Corrente, and Salvatore Greco. Stochastic multiobjective acceptability analysis for the choquet integral preference model and the scale construction problem. *European Journal of Operational Research*, 240(1):172–182, 2015. (Cited on page 28.)

Barry C Arnold, Narayanaswamy Balakrishnan, and Haikady Navada Nagaraja. *A first course in order statistics*. SIAM, 2008. (Cited on page 235.)

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950. (Cited on pages 150 and 154.)

Kenneth J Arrow. Social choice and individual values. *Cowles Commission Mongr. No. 12.*, 1951. (Cited on page 8.)

Kenneth J Arrow. The theory of risk aversion. *Essays in the theory of risk-bearing*, pages 90–120, 1971. (Cited on page 62.)

Nicolas Atienza, Roman Bresson, Cyriaque Rousselot, Philippe Caillou, Johanne Cohen, Christophe Labreuche, and Michele Sebag. Cutting the black box: Conceptual interpretation of a deep neural net with multi-modal embeddings and multi-criteria decision aid. In *Proceedings of the Thirty Third International Conference on International Joint Conferences on Artificial Intelligence*, page To appear, 2024. (Cited on pages 3 and 52.)

Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Ruth Urner. Efficient learning of linear separators under bounded noise. In *Conference on Learning Theory*, pages 167–190. PMLR, 2015. (Cited on page 223.)

Ismaïl Baaj. On learning capacities of sugeno integrals with systems of fuzzy relational equations. *arXiv preprint arXiv:2408.07768*, 2024. (Cited on page 228.)

F Bacchus and A Grove. Graphical models for preference and utility. In *UAI'95*, 1995. (Cited on pages 1 and 142.)

Francis Bach. Exploring large feature spaces with hierarchical multiple kernel learning. *Advances in neural information processing systems*, 21, 2008. (Cited on page 164.)

Francis Bach, Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, et al. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106, 2012. (Cited on pages 3, 38, 42, 45, 111, and 159.)

Francis R Bach. The "eta-trick" or the effectiveness of reweighted least-squares, 2019. Online ; Available at? ; accessed 27 September 2024. (Cited on page 112.)

Francis R Bach, Gert RG Lanckriet, and Michael I Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, page 6, 2004. (Cited on pages 150, 158, and 159.)

Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 65–72, 2006. (Cited on pages 181 and 182.)

Carlos A Bana e Costa and Jean-Claude Vansnick. A theoretical framework for measuring attractiveness by a categorical based evaluation technique (MACBETH). In *Multicriteria Analysis: Proceedings of the XIth International Conference on MCDM*, pages 15–24, 1997. (Cited on pages 28 and 59.)

Pathikrit Basu and Federico Echenique. On the falsifiability and learnability of decision theories. *Theoretical Economics*, 15(4):1279–1305, 2020. (Cited on pages 195 and 230.)

Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer Publishing Company, Incorporated, 1st edition, 2011. ISBN 1441994661. (Cited on page 39.)

Amir Beck. On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes. *SIAM Journal on Optimization*, 25(1):185–209, 2015. (Cited on pages 3, 107, 111, 112, 113, and 114.)

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003. (Cited on pages 44, 202, and 203.)

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009. (Cited on pages 3 and 45.)

Khaled Belahcene, Nataliya Sokolovska, Yann Chevaleyre, and Jean-Daniel Zucker. Learning interpretable models using soft integrity constraints. In *Asian Conference on Machine Learning*, pages 529–544. PMLR, 2020. (Cited on page 42.)

G Beliakov, Marek Gągolewski, and S James. Dc optimization for constructing discrete sugeno integrals and learning nonadditive measures. *Optimization*, 69(12):2515–2534, 2020. (Cited on page 52.)

Gleb Beliakov and Dmitriy Divakov. On representation of fuzzy measures for learning choquet and sugeno integrals. *Knowledge-Based Systems*, 189:105134, 2020. (Cited on page 228.)

Gleb Beliakov and Jian-Zhang Wu. Learning fuzzy measures from data: simplifications and optimisation strategies. *Information Sciences*, 494:100–113, 2019a. (Cited on page 198.)

Gleb Beliakov and Jian-Zhang Wu. Learning fuzzy measures from data: simplifications and optimisation strategies. *Information Sciences*, 494:100–113, 2019b. (Cited on pages 30, 31, and 52.)

Gleb Beliakov and Jian-Zhang Wu. Learning k-maxitive fuzzy measures from data by mixed integer programming. *Fuzzy Sets and Systems*, 412:41–52, 2021. (Cited on pages 2, 30, and 31.)

Nawal Benabbou, Patrice Perny, and Paolo Viappiani. Incremental elicitation of Choquet capacities for multicriteria choice, ranking and sorting problems. *Artificial Intelligence*, 246:152–180, 2017a. (Cited on pages 30, 176, 177, and 198.)

Nawal Benabbou, Patrice Perny, and Paolo Viappiani. Incremental elicitation of choquet capacities for multicriteria choice, ranking and sorting problems. *Artificial Intelligence*, 246:152–180, 2017b. (Cited on page 230.)

Nawal Benabbou, Cassandre Leroy, and Thibaut Lust. An interactive regret-based genetic algorithm for solving multi-objective combinatorial optimization problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2335–2342, 2020. (Cited on page 176.)

Kristin P Bennett and Emilio Parrado-Hernández. The interplay of optimization and machine learning research. *The Journal of Machine Learning Research*, 7:1265–1281, 2006. (Cited on pages 3 and 44.)

Dimitri Bertsekas. *Convex optimization algorithms*. Athena Scientific, 2015. (Cited on page 43.)

Dimitri P Bertsekas. Constrained optimization and lagrange multiplier methods. *Computer Science and Applied Mathematics*, 1982. (Cited on page 212.)

Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997. (Cited on page 118.)

Dimitris Bertsimas and Bart Van Parys. Sparse high-dimensional regression: Exact scalable algorithms and phase transitions. *The Annals of Statistics*, 48(1):pp. 300–323, 2020. ISSN 00905364, 21688966. (Cited on page 229.)

Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. 2016. (Cited on page 229.)

Damien Bigot, Hélene Fargier, Jérôme Mengin, and Bruno Zanuttini. Using and learning gai-decompositions for representing ordinal rankings. In *ECAI'2012 workshop on Preference Learning (PL 2012)*, pages 5–10. Fürnkranz Johannes Hüllermeier Eyke, 2012. (Cited on pages 3, 53, and 142.)

José M Bioucas-Dias and Mário AT Figueiredo. Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing. In *2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, pages 1–4. IEEE, 2010. (Cited on page 212.)

Michael J Black and Anand Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International journal of computer vision*, 19(1):57–91, 1996. (Cited on page 111.)

Pavlo Blavatskyy. Error propagation in the elicitation of utility and probability weighting functions. *Theory and Decision*, 60(2):315–334, 2006. (Cited on pages 27, 59, and 67.)

Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992. (Cited on pages 38 and 150.)

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, pages 177–186. Springer, 2010. (Cited on page 44.)

Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. *Advances in neural information processing systems*, 20, 2007. (Cited on pages 3 and 44.)

Léon Bottou and Yann Cun. Large scale online learning. *Advances in neural information processing systems*, 16, 2003. (Cited on page 203.)

Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018. doi: 10.1137/16M1080173. (Cited on pages 3 and 203.)

Nadjet Bourdache, Patrice Perny, and Olivier Spanjaard. Incremental elicitation of rank-dependent aggregation functions based on Bayesian linear regression. In *IJCAI-19-Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 2023–2029, 2019a. (Cited on pages 30, 176, 177, and 195.)

Nadjet Bourdache, Patrice Perny, and Olivier Spanjaard. Incremental elicitation of rank-dependent aggregation functions based on bayesian linear regression. In *IJCAI-19-twenty-eighth international joint conference on artificial intelligence*, pages 2023–2029. International Joint Conferences on Artificial Intelligence Organization, 2019b. (Cited on page 230.)

Cécile Bourreau-Dubois, Myriam Doriat-Duban, Agnès Gramain, Bruno Jeandidier, Julie Mansuy, and Jean-Claude Ray. Analyses quantitatives de décisions de justice en matière de prestation compensatoire dans une perspective de justice prédictive. Technical report, Bureau d'Economie Théorique et Appliquée, UDS, Strasbourg, 2022. (Cited on pages 135, 136, and 238.)

Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In *Summer school on machine learning*, pages 169–207. Springer, 2003. (Cited on page 33.)

Craig Boutilier, Ronen Brafman, Chris Geib, and David Poole. A constraint-based approach to preference elicitation and decision making. In *AAAI Spring Symposium on qualitative decision theory*, pages 19–28. Citeseer, 1997. (Cited on page 26.)

Craig Boutilier, Ronen I Brafman, Holger H Hoos, and David Poole. Reasoning with conditional ceteris paribus preference statements. In *UAI*, volume 99, pages 71–80, 1999. (Cited on page 1.)

Craig Boutilier, Relu Patrascu, Pascal Poupart, and Dale Schuurmans. Constraint-based optimization and utility elicitation using the minimax decision criterion. *Artificial Intelligence*, 170(8-9):686–713, 2006. (Cited on pages 30, 176, and 177.)

Denis Bouyssou. *Evaluation and decision models: a critical perspective*, volume 32. Springer Science & Business Media, 2000. (Cited on page 27.)

Denis Bouyssou and Marc Pirlot. Conjoint measurement tools for mcdm: A brief introduction. *Multiple criteria decision analysis: State of the art surveys*, pages 97–151, 2016. (Cited on page 14.)

Denis Bouyssou and Philippe Vincke. Binary relations and preference modeling. *Decision-making Process: Concepts and Methods*, pages 49–84, 2009. (Cited on pages 6, 9, and 10.)

Denis Bouyssou, Didier Dubois, Henri Prade, and Marc Pirlot. *Decision making process: Concepts and methods*. John Wiley & Sons, 2013. (Cited on page 14.)

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004. (Cited on pages 36, 37, 117, and 118.)

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011. (Cited on pages 46, 198, 211, 212, and 213.)

Ronen Brafman and Carmel Domshlak. Preference handling-an introductory tutorial. *AI magazine*, 30(1):58–58, 2009. (Cited on page 6.)

Ronen Brafman and Yagil Engel. Decomposed utility functions and graphical models for reasoning about preferences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pages 267–272, 2010. (Cited on page 142.)

D Braziunas and C Boutilier. Local utility elicitation in GAI models. In *UAI'05*, 2005. (Cited on pages 2, 31, 142, and 149.)

Darius Braziunas. *Decision-theoretic Elicitation of Generalized Additive Utilities*. PhD thesis, University of Toronto, 2012. (Cited on pages 31, 142, and 149.)

Darius Braziunas and Craig Boutilier. Elicitation of factored utilities. *AI Magazine*, 29 (4):79–79, 2008. (Cited on pages 6 and 142.)

Lev M Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967. (Cited on page 202.)

Leo Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231, 2001. (Cited on page 71.)

Roman Bresson. *Neural learning and validation of hierarchical multi-criteria decision aiding models with interacting criteria*. PhD thesis, Université Paris-Saclay, 2022. (Cited on pages 52, 59, and 103.)

Roman Bresson, Johanne Cohen, Eyke Hüllermeier, Christophe Labreuche, and Michèle Sebag. Neural representation and learning of hierarchical 2-additive Choquet integrals. In *IJCAI*, pages 1984–1991, 2020. (Cited on page 198.)

Roman Bresson, Johanne Cohen, Eyke Hüllermeier, Christophe Labreuche, and Michele Sebag. Neural representation and learning of hierarchical 2-additive choquet integrals. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 1984–1991, 2021. (Cited on pages 3, 52, 103, and 106.)

Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015. (Cited on pages 43 and 47.)

Serhat S Bucak, Rong Jin, and Anil K Jain. Multiple kernel learning for visual object recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1354–1369, 2013. (Cited on page 158.)

Róbert Busa-Fekete and Eyke Hüllermeier. A survey of preference-based online learning with bandit algorithms. In *Algorithmic Learning Theory: 25th International Conference, ALT 2014, Bled, Slovenia, October 8-10, 2014. Proceedings 25*, pages 18–39. Springer, 2014. (Cited on page 32.)

Richard H Byrd, Samantha L Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-newton method for large-scale optimization. *SIAM Journal on Optimization*, 26 (2):1008–1031, 2016. (Cited on page 45.)

Davide Cacciarelli and Murat Kulahci. Active learning for data streams: a survey. *Machine Learning*, 113(1):185–239, 2024. (Cited on page 181.)

Augustin Cauchy et al. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847. (Cited on page 43.)

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games.* Cambridge university press, 2006. (Cited on page 203.)

Urszula Chajewska, Daphne Koller, and Ronald Parr. Making rational decisions using adaptive utility elicitation. In *AAAI/IAAI*, pages 363–369, 2000. (Cited on pages 30, 176, and 195.)

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. (Cited on pages 136 and 158.)

Olivier Chapelle and Zaid Harchaoui. A machine learning approach to conjoint analysis. *Advances in neural information processing systems*, 17, 2004. (Cited on pages 120 and 160.)

Alain Chateauneuf and Jean-Yves Jaffray. Some characterizations of lower probabilities and other monotone capacities through the use of möbius inversion. *Mathematical social sciences*, 17(3):263–283, 1989. (Cited on pages 16 and 107.)

Alain Chateauneuf and Jean-Marc Tallon. Diversification, convex preferences and non-empty core in the Choquet expected utility model. *Econ. Theory*, 19(3):509–523, 2002a. (Cited on pages 63 and 209.)

Alain Chateauneuf and Jean-Marc Tallon. Diversification, convex preferences and non-empty core in the choquet expected utility model. *Economic Theory*, 19:509–523, 2002b. (Cited on page 20.)

Lin Chen, Christopher Harshaw, Hamed Hassani, and Amin Karbasi. Projection-free online optimization with stochastic gradient: From convexity to submodularity. In *International Conference on Machine Learning*, pages 814–823. PMLR, 2018. (Cited on page 211.)

Wei Chen, Tie-Yan Liu, Yanyan Lan, Zhi-Ming Ma, and Hang Li. Ranking measures and loss functions in learning to rank. *Advances in Neural Information Processing Systems*, 22, 2009. (Cited on page 167.)

Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet. Preference handling in combinatorial domains: From ai to social choice. *AI magazine*, 29(4):37–37, 2008. (Cited on page 6.)

Yann Chevaleyre, Frédéric Koriche, Jérôme Lang, Jérôme Mengin, and Bruno Zanuttini. Learning ordinal preferences on multiattribute domains: The case of cp-nets. In *Preference learning*, pages 273–296. Springer, 2010. (Cited on page 6.)

Gustave Choquet. Theory of capacities. In *Annales de l'institut Fourier*, volume 5, pages 131–295, 1954. (Cited on page 14.)

David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine learning*, 15:201–221, 1994. (Cited on page 181.)

Elías F Combarro and Pedro Miranda. Identification of fuzzy measures from sample data with genetic algorithms. *Computers & Operations Research*, 33(10):3046–3066, 2006. (Cited on page 52.)

Salvatore Corrente, Salvatore Greco, Miłosz Kadziński, and Roman Słowiński. Robust ordinal regression in preference learning and ranking. *Machine Learning*, 93:381–422, 2013. (Cited on page 30.)

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20: 273–297, 1995. (Cited on page 118.)

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. L 2 regularization for learning kernels. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 109–116, 2009. (Cited on page 159.)

Corinna Cortes, Giulia DeSalvo, Mehryar Mohri, Ningshan Zhang, and Claudio Gentile. Active learning with disagreement graphs. In *International Conference on Machine Learning*, pages 1379–1387. PMLR, 2019. (Cited on page 181.)

Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods.* Cambridge university press, 2000. (Cited on pages 118 and 119.)

Sébastien Da Veiga, Fabrice Gamboa, Bertrand Iooss, and Clémentine Prieur. *Basics and trends in sensitivity analysis: Theory and practice in R.* SIAM, 2021. (Cited on page 145.)

George B Dantzig. Maximization of a linear function of variables subject to linear inequalities. *Activity analysis of production and allocation*, 13:339–347, 1951. (Cited on page 47.)

Sanjoy Dasgupta. Two faces of active learning. *Theoretical computer science*, 412(19): 1767–1781, 2011. (Cited on page 181.)

Sanjoy Dasgupta, Daniel J Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. *Advances in neural information processing systems*, 20, 2007. (Cited on pages 182, 184, and 185.)

Ingrid Daubechies, Ronald DeVore, Massimo Fornasier, and C Sinan Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 63(1):1–38, 2010. (Cited on pages 46, 107, and 111.)

C De Boor. A practical guide to splines. *Springer-Verlag google schola*, 2:4135–4195, 1978. (Cited on page 67.)

Henrique Evangelista de Oliveira, Leonardo Tomazeli Duarte, and João Marcos Travassos Romano. Identification of the Choquet integral parameters in the interaction index domain by means of sparse modeling. *Expert Systems with Applications*, 187, 2022. (Cited on pages 3, 53, 58, and 106.)

Gerard Debreu et al. Representation of a preference ordering by a numerical function. *Decision processes*, 3:159–165, 1954. (Cited on page 10.)

AP Dempster. Upper and lower probabilities induced by a multivalued mapping. *Ann. Math. Statist.*, 38(6):325–339, 1967. (Cited on page 78.)

Wei Deng and Wotao Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. *Journal of Scientific Computing*, 66:889–916, 2016. (Cited on page 217.)

Giulia DeSalvo, Claudio Gentile, and Tobias Sommer Thune. Online active learning with surrogate loss functions. *Advances in neural information processing systems*, 34: 22877–22889, 2021. (Cited on pages 181 and 196.)

Luc Devroye, László Györfi, and Gábor Lugosi. A probabilistic theory of pattern recognition. *Stochastic Modelling and Applied Probability*, 1996. (Cited on page 33.)

Carmel Domshlak and Thorsten Joachims. Unstructuring user preferences: Efficient non-parametric utility revelation. *arXiv preprint arXiv:1207.1390*, 2012. (Cited on pages 120, 123, and 160.)

Carmel Domshlak, Eyke Hüllermeier, Souhila Kaci, and Henri Prade. Preferences in ai: An overview. *Artificial Intelligence*, 175(7-8):1037–1052, 2011a. (Cited on pages 32 and 198.)

Carmel Domshlak, Eyke Hüllermeier, Souhila Kaci, and Henri Prade. Preferences in ai: An overview. *Artificial Intelligence*, 175(7-8):1037–1052, 2011b. (Cited on page 6.)

Haris Doukas and Alexandros Nikas. Decision support models in climate policy. *European Journal of Operational Research*, 280(1):1–24, 2020. (Cited on page 1.)

Didier Dubois, Henri Prade, and Régis Sabbadin. Qualitative decision theory with sugeno integrals. In *14th Conference on Uncertainty in Artificial Intelligence (UAI 1998)*, pages 121–128. Morgan Kaufmann, 1998. (Cited on page 24.)

Didier Dubois, Helene Fargier, Patrice Perny, and Henri Prade. A characterization of generalized concordance rules in multicriteria decision making. *International Journal of Intelligent Systems*, 18(7):751–774, 2003. (Cited on page 228.)

J. P. Dubus, C. Gonzales, and P. Perny. Fast recommendations using gai models. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, volume 9, pages 1896–1901, July 2009. (Cited on page 142.)

John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *The Journal of Machine Learning Research*, 10:2899–2934, 2009. (Cited on pages 203 and 221.)

John C Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite objective mirror descent. In *COLT*, volume 10, pages 14–26. Citeseer, 2010. (Cited on page 203.)

Nicolas Durrande. *Étude de classes de noyaux adaptées à la simplification et à l'interprétation des modèles d'approximation. Une approche fonctionnelle et probabiliste.* PhD thesis, Saint-Etienne, EMSE, 2011. (Cited on page 164.)

Nicolas Durrande, David Ginsbourger, Olivier Roustant, and Laurent Carraro. Anova kernels and rkhs of zero mean functions for model-based sensitivity analysis. *Journal of Multivariate Analysis*, 115:57–67, 2013. (Cited on page 163.)

James S Dyer. Maut—multiattribute utility theory. *Multiple criteria decision analysis: state of the art surveys*, pages 265–292, 2005. (Cited on pages 7 and 72.)

James S Dyer and Rakesh K Sarin. Measurable multiattribute value functions. *Operations research*, 27(4):810–822, 1979. (Cited on page 21.)

Daniel Ellsberg. Risk, ambiguity, and the Savage axioms. *The quarterly journal of econ.*, pages 643–669, 1961. (Cited on page 62.)

Yagil Engel and Michael P Wellman. Multiattribute auctions based on generalized additive independence. *Journal of Artificial Intelligence Research*, 37:479–525, 2010. (Cited on page 142.)

Guillaume Escamocher, Samira Pourkhajouei, Federico Toffano, Paolo Viappiani, and Nic Wilson. Interactive preference elicitation under noisy preference models: An efficient non-bayesian approach. *International Journal of Approximate Reasoning*, 178:109333, 2025. (Cited on page 193.)

Theodoros Evgeniou, Constantinos Boussios, and Giorgos Zacharia. Generalized robust conjoint estimation. *Marketing Science*, 24(3):415–429, 2005. (Cited on page 167.)

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020. (Cited on page 230.)

Rong-En Fan, Pai-Hsuen Chen, Chih-Jen Lin, and Thorsten Joachims. Working set selection using second order information for training support vector machines. *Journal of machine learning research*, 6(12), 2005. (Cited on pages 47 and 158.)

P. C. Fishburn. *Utility Theory for Decision Making.* Wiley, 1970. (Cited on pages 1, 25, and 142.)

Peter C Fishburn. Interdependence and additivity in multivariate, unidimensional expected utility theory. *International Economic Review*, 8(3):335–342, 1967. (Cited on page 149.)

Peter C Fishburn, Peter C Fishburn, et al. *Utility theory for decision making.* Krieger NY, 1979. (Cited on page 10.)

Pedro A Forero, Alfonso Cano, and Georgios B Giannakis. Consensus-based distributed linear support vector machines. In *Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks*, pages 35–46, 2010. (Cited on page 212.)

Marguerite Frank, Philip Wolfe, et al. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956. (Cited on pages 44 and 211.)

Jerome H Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33:1–22, 2010. (Cited on pages 3 and 46.)

Fabian Fumagalli, Maximilian Muschalik, Patrick Kolpaczki, Eyke Hüllermeier, and Barbara Hammer. Kernelshap-iq: Weighted least-square optimization for shapley interactions. *arXiv preprint arXiv:2405.10852*, 2024. (Cited on page 231.)

Johannes Fürnkranz and Eyke Hüllermeier. Preference learning and ranking by pairwise comparison. In *Preference learning*, pages 65–82. Springer, 2010a. (Cited on page 198.)

Johannes Fürnkranz and Eyke Hüllermeier. Preference learning and ranking by pairwise comparison. In *Preference learning*, pages 65–82. Springer, 2010b. (Cited on pages 3 and 32.)

Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & mathematics with applications*, 2(1):17–40, 1976. (Cited on page 212.)

Marek Gagolewski, Simon James, and Gleb Beliakov. Supervised learning to aggregate data with the Sugeno integral. *IEEE Transactions on Fuzzy Systems*, 27(4):810 – 815, 2019a. (Cited on page 52.)

Marek Gagolewski, Simon James, and Gleb Beliakov. Supervised learning to aggregate data with the sugeno integral. *IEEE Transactions on Fuzzy Systems*, 27(4):810–815, 2019b. (Cited on page 228.)

Lucie Galand and Brice Mayag. A heuristic approach to test the compatibility of a preference information with a Choquet integral model. In *ADT*, pages 65 – 80, 2017a. (Cited on page 29.)

Lucie Galand and Brice Mayag. A heuristic approach to test the compatibility of a preference information with a choquet integral model. In *Algorithmic Decision Theory - 5th International Conference, ADT*, pages 65–80, 2017b. (Cited on pages 31 and 106.)

Xiaoli Gao and Jian Huang. Asymptotic analysis of high-dimensional LAD regression with lasso. *Statistica Sinica*, pages 1485–1506, 2010. (Cited on pages 82, 83, and 84.)

Hugo Gilbert, Mohamed Ouaguenouni, Meltem Öztürk, and Olivier Spanjaard. Robust ordinal regression for subsets comparisons with interactions. *European Journal of Operational Research*, 320(1):146–159, 2025. (Cited on page 30.)

Roland Glowinski and Americo Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 9(R2):41–76, 1975. (Cited on pages 46, 198, and 212.)

Donald Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of computation*, 24(109):23–26, 1970. (Cited on page 45.)

Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *The Journal of Machine Learning Research*, 12:2211–2268, 2011. (Cited on pages 143, 158, and 159.)

C Gonzales and P Perny. GAI networks for utility elicitation. In *KR'04*, pages 224–234, 2004. (Cited on pages 31 and 142.)

Christophe Gonzales and Patrice Perny. Gai networks for decision making under certainty. In *IJCAI'05–Workshop on Advances in Preference Handling*, pages 100–105, 2005. (Cited on page 31.)

Christophe Gonzales and Patrice Perny. Decision under uncertainty. *A Guided Tour of Artificial Intelligence Research: Volume I: Knowledge Representation, Reasoning and Learning*, pages 549–586, 2020. (Cited on page 61.)

Christophe Gonzales, Patrice Perny, and Sergio Queiroz. Gai-networks: Optimization, ranking and collective choice in combinatorial domains. *Foundations of computing and decision sciences*, 33(1):3–24, 2008. (Cited on page 142.)

Bénédicte Goujon. Preference learning for object ranking and classification with fixed-point algorithm, 2018. (Cited on page 28.)

Bénédicte Goujon and Christophe Labreuche. Holistic preference learning with the Choquet integral. In *EUSFLAT*, 2013. (Cited on page 28.)

Michel Grabisch. The application of fuzzy integrals in multicriteria decision making. *European Journal of Operational Research*, 89(3):445–456, 1996. (Cited on pages 1 and 14.)

Michel Grabisch. K-order additive discrete fuzzy measures and their representation. *Fuzzy sets and systems*, 92(2):167–189, 1997a. (Cited on page 228.)

Michel Grabisch. K-order additive discrete fuzzy measures and their representation. *Fuzzy sets and systems*, 92(2):167–189, 1997b. (Cited on pages 17 and 79.)

Michel Grabisch. *Aggregation functions*, volume 127. Cambridge University Press, 2009. (Cited on page 11.)

Michel Grabisch. *Set functions, games and capacities in decision making*. Springer, 2016a. (Cited on page 209.)

Michel Grabisch. Decision with multiple criteria. *Set Functions, Games and Capacities in Decision Making*, pages 325–375, 2016b. (Cited on pages 7, 28, and 72.)

Michel Grabisch. Dempster-shafer and possibility theory. *Set Functions, Games and Capacities in Decision Making*, pages 377–437, 2016c. (Cited on page 78.)

Michel Grabisch and Christophe Labreuche. A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid. *Annals of Operations Research*, 175(1): 247–286, 2010. (Cited on pages 24, 30, 79, and 106.)

Michel Grabisch and Jean-Marie Nicolas. Classification by fuzzy integral: Performance and tests. *Fuzzy sets and systems*, 65(2-3):255–271, 1994. (Cited on page 51.)

Michel Grabisch, Jean-Luc Marichal, and Marc Roubens. Equivalent representations of a set function with application to game theory and multicriteria decision making. In *The Fall 1998 Meeting of the Institute for Operations Research and the Management Sciences (INFORMS), Seattle, USA, Oct. 25-28, 1998*, 1998. (Cited on page 17.)

Michel Grabisch, Ivan Kojadinovic, and Patrick Meyer. A review of methods for capacity identification in Choquet integral based multi-attribute utility theory: Applications of the Kappalab R package. *European Journal of Operational Research*, 186(2):766–785, 2008. (Cited on pages 2, 30, 31, 106, and 198.)

Michel Grabisch, Jean-Luc Marichal, Radko Mesiar, and Endre Pap. *Aggregation functions*, volume 127. Cambridge University Press, 2009. (Cited on pages 14 and 108.)

Michel Grabisch, Christophe Labreuche, and Mustapha Ridaoui. Well-formed decompositions of generalized additive independence models. *Annals of Operations Research*, 312(2):827–852, 2022. (Cited on pages 2, 31, 142, 144, 145, and 174.)

Michel Grabisch, Christophe Labreuche, and Peiqi Sun. An approximation algorithm for random generation of capacities. *Order*, pages 1–26, 2023. (Cited on page 90.)

Michel Grabisch et al. *Set functions, games and capacities in decision making*, volume 46. Springer, 2016. (Cited on pages 11, 21, 22, and 191.)

Yves Grandvalet. Least absolute shrinkage is equivalent to quadratic penalization. In *International Conference on Artificial Neural Networks*, pages 201–206. Springer, 1998. (Cited on page 111.)

Salvatore Greco, Vincent Mousseau, and Roman Słowiński. Ordinal regression revisited: multiple criteria ranking using a set of additive value functions. *European Journal of Operational Research*, 191(2):416–436, 2008. (Cited on page 30.)

Michael Griebel and Markus Holtz. Dimension-wise integration of high-dimensional functions with applications to finance. *Journal of Complexity*, 26(5):455–489, 2010. (Cited on pages 145 and 148.)

Chong Gu. Smoothing spline anova models. *Springer Series in Statistics*, 2002. (Cited on page 161.)

Chong Gu. *Smoothing spline ANOVA models*, volume 297. Springer, 2013. (Cited on page 162.)

Chong Gu. Smoothing spline anova models: R package gss. *Journal of Statistical Software*, 58:1–25, 2014. (Cited on page 161.)

Yanfeng Gu, Jocelyn Chanussot, Xiuping Jia, and Jon Atli Benediktsson. Multiple kernel learning for hyperspectral image classification: A review. *IEEE Transactions on Geoscience and Remote Sensing*, 55(11):6547–6565, 2017. (Cited on page 158.)

Steve R. Gunn and Jaz S. Kandola. Structural modelling with sparse kernels. *Machine learning*, 48:137–163, 2002. (Cited on pages 163 and 164.)

Theo Guyard, Cédric Herzet, Clément Elvira, and Ayse-Nur Arslan. A new branch-and-bound pruning framework for *ell*_0-regularized problems. In *International Conference on Machine Learning*, pages 48077–48096. PMLR, 2024. (Cited on page 229.)

Steve Hanneke et al. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309, 2014. (Cited on pages 181, 229, and 230.)

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009. (Cited on page 67.)

Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical learning with sparsity: the Lasso and Generalizations*. CRC Press, 2015a. (Cited on pages 3 and 40.)

Trevor Hastie, Robert Tibshirani, and Martin Wainwright. Statistical learning with sparsity. *Monographs on Statistics and Applied Probability*, 143, 2015b. (Cited on pages 45, 83, and 84.)

Timothy C Havens and Anthony J Pinar. Generating random fuzzy (capacity) measures for data fusion simulations. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, 2017. (Cited on page 90.)

Elad Hazan and Satyen Kale. Projection-free online learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1843–1850, 2012. (Cited on page 211.)

Elad Hazan and Edgar Minasyan. Faster projection-free online learning. In *Conference on Learning Theory*, pages 1877–1893. PMLR, 2020. (Cited on page 211.)

Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016. (Cited on pages 200 and 202.)

Bingsheng He and Xiaoming Yuan. On the o(1/n) convergence rate of the douglas–rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2): 700–709, 2012. (Cited on pages 213, 217, and 218.)

Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Large margin rank boundaries for ordinal regression. 2000. (Cited on page 167.)

Margot Herin, Patrice Perny, and Nataliya Sokolovska. Learning sparse representations of preferences within Choquet expected utility theory. In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022a. (Cited on pages 4, 57, and 106.)

Margot Herin, Patrice Perny, and Nataliya Sokolovska. Learning Utilities and Sparse Representations of Capacities for Multicriteria Decision Making with the Bipolar Choquet Integral. In *Multidisciplinary Workshop on Advances in Preference Handling, IJCAI*, Vienne, Austria, July 2022b. URL https://hal.science/hal-03780464. (Cited on pages 4 and 57.)

Margot Herin, Patrice Perny, and Nataliya Sokolovska. A Dual Approach for Learning Sparse Representations of Choquet Integrals. In *DA2PL From Multiple-Criteria Decision Aid to Preference Learning*, Compiègne, France, November 2022c. URL https://hal.science/hal-03978343. (Cited on pages 4 and 105.)

Margot Herin, Patrice Perny, and Nataliya Sokolovska. Learning preference models with sparse interactions of criteria. In *Proc. of IJCAI*, pages 3786–3794, 2023a. (Cited on page 198.)

Margot Herin, Patrice Perny, and Nataliya Sokolovska. Learning preference models with sparse interactions of criteria. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 3786–3794, 2023b. (Cited on pages 4 and 105.)

Margot Herin, Patrice Perny, and Nataliya Sokolovska. Noise-tolerant active preference learning for multicriteria choice problems. In *International Conference on Algorithmic Decision Theory*, pages 191–206. Springer, 2024a. (Cited on pages 4 and 175.)

Margot Herin, Patrice Perny, and Nataliya Sokolovska. Learning GAI-decomposable Utility Models for Multiattribute Decision Making. In *The 38th Annual AAAI Conference on Artificial Intelligence (AAAI 2024)*, Vancouver, Canada, February 2024b. URL https://hal.science/hal-04424705. (Cited on pages 4 and 141.)

Margot Herin, Patrice Perny, and Nataliya Sokolovska. Learning preference representations based on choquet integrals for multicriteria decision making. *Annals of Mathematics and Artificial Intelligence*, pages 1–34, 2024c. (Cited on pages 4, 57, and 106.)

Margot Herin, Patrice Perny, and Nataliya Sokolovska. Online learning of capacity-based preference models. In *International Joint Conference on Artificial Intelligence (IJCAI) 2024*, pages 7118–7126, 2024d. (Cited on pages 4 and 197.)

Greg Hines and Kate Larson. Preference elicitation for risky prospects. In *AAMAS*, pages 889–896, 2010. (Cited on page 64.)

Wassily Hoeffding. A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics*, 19(3):293–325, 1948. (Cited on page 145.)

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. (Cited on page 41.)

Steven CH Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. Online learning: A comprehensive survey. *Neurocomputing*, 459:249–289, 2021. (Cited on pages 198, 199, 204, and 206.)

Saghar Hosseini, Airlie Chapman, and Mehran Mesbahi. Online distributed admm via dual averaging. In *53rd IEEE Conference on Decision and Control*, pages 904–909. IEEE, 2014. (Cited on pages 211 and 215.)

Junzhou Huang and Tong Zhang. The benefit of group sparsity. *Annals of Statistics*, 38 (4):1978–2004, 2010. (Cited on page 42.)

Eyke Hüllermeier and Ali Fallah Tehrani. On the vc-dimension of the choquet integral. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 42–50. Springer, 2012. (Cited on pages 195 and 230.)

Eyke Hüllermeier and Johannes Fürnkranz. Preference learning and ranking, 2013. (Cited on page 6.)

Eyke Hüllermeier and Roman Słowiński. Preference learning and multiple criteria decision aiding: differences, commonalities, and synergies–part i. *4OR*, 22(2):179–209, 2024a. (Cited on page 32.)

Eyke Hüllermeier and Roman Słowiński. Preference learning and multiple criteria decision aiding: differences, commonalities, and synergies—part ii. *4OR*, 22(3):313–349, 2024b. (Cited on pages 3 and 32.)

Eyke Hüllermeier and Ali Fallah Tehrani. Efficient learning of classifiers based on the 2-additive choquet integral. In *Computational Intelligence in Intelligent Data Analysis*, pages 17–29, 2013. (Cited on pages 79 and 106.)

Leonid Hurwicz. The generalized Bayes minimax principle: a criterion for decision making under uncertainty. *Cowles Comm. Discuss. Paper Stat*, 335:1950, 1951. (Cited on pages 89 and 133.)

Eric Jacquet-Lagreze and Jean Siskos. Assessing a set of additive utility functions for multicriteria decision-making, the uta method. *European journal of operational research*, 10(2):151–164, 1982. (Cited on page 27.)

Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International conference on machine learning*, pages 427–435. PMLR, 2013. (Cited on page 44.)

Bruno Jeandidier. L'hétérogénéité des décisions de justice réduit leur prévisibilité. In Isabelle Sayn, editor, *Justice et numérique – Quels (r)apports ?*, pages 65–83. Presses Universitaires Savoie Mont Blanc, 2024. (Cited on pages 135, 136, 137, and 238.)

Bruno Jeandidier, Jean-Claude Ray, and Julie Mansuy. Analyses quantitatives de décisions de justice en matière de prestation compensatoire (pc) dans une perspective de justice prédictive. 2020. (Cited on pages 135, 136, and 238.)

Rodolphe Jenatton, Jim Huang, and Cédric Archambeau. Adaptive algorithms for online convex optimization with long-term constraints. In *International Conference on Machine Learning*, pages 402–411. PMLR, 2016. (Cited on page 211.)

Michael Jünger, Gerhard Reinelt, and Stefan Thienel. Practical problem solving with cutting plane algorithms in combinatorial optimization. In *Combinatorial Optimization*, 1993. (Cited on page 128.)

Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–292, 1979. (Cited on pages 28, 59, and 225.)

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021. (Cited on page 230.)

Siva K Kakula, Anthony J Pinar, Timothy C Havens, and Derek T Anderson. Choquet integral ridge regression. In *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE, 2020a. (Cited on pages 3 and 52.)

Siva K Kakula, Anthony J Pinar, Timothy C Havens, and Derek T Anderson. Online learning of the fuzzy choquet integral. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 608–614. IEEE, 2020b. (Cited on pages 52, 53, 58, and 198.)

Leonid Vital'evich Kantorovich. On newton's method. *Trudy Matematicheskogo Instituta imeni VA Steklova*, 28:104–144, 1949. (Cited on page 44.)

Ioannis Karatzas and Steven Shreve. *Brownian motion and stochastic calculus*, volume 113. Springer Science & Business Media, 1991. (Cited on page 163.)

Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311, 1984. (Cited on page 46.)

Ralph L Keeney and Howard Raiffa. *Decision analysis with multiple conflicting objectives, preferences and value tradeoffs.* Wiley & Sons, New York, 1976. (Cited on pages 1, 7, 21, and 72.)

Ralph L Keeney, Howard Raiffa, and Richard F Meyer. *Decisions with multiple objectives: preferences and value trade-offs.* Cambridge university press, 1993. (Cited on pages 28 and 103.)

George Kimeldorf and Grace Wahba. Some results on tchebycheffian spline functions. *Journal of mathematical analysis and applications*, 33(1):82–95, 1971. (Cited on pages 150, 151, and 152.)

Marius Kloft, Ulf Brefeld, Pavel Laskov, Klaus-Robert Müller, Alexander Zien, and Sören Sonnenburg. Efficient and accurate lp-norm multiple kernel learning. *Advances in neural information processing systems*, 22, 2009. (Cited on page 159.)

Marius Kloft, Ulf Brefeld, Sören Sonnenburg, and Alexander Zien. Lp-norm multiple kernel learning. *The Journal of Machine Learning Research*, 12:953–997, 2011. (Cited on pages 159, 160, and 174.)

Veronika Köbberling and Peter P Wakker. Preference foundations for nonexpected utility: A generalized and simplified technique. *Mathematics of Operations Research*, 28(3): 395–423, 2003. (Cited on page 20.)

Anna Kolesarova, Andrea Stupňanová, and Juliana Beganova. Aggregation-based extensions of fuzzy measures. *Fuzzy Sets and Systems*, 194:1–14, 2012. (Cited on page 109.)

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009. (Cited on page 142.)

David H Krantz and Amos Tversky. Conjoint-measurement analysis of composition rules in psychology. *Psychological Review*, 78(2):151, 1971. (Cited on pages 11, 27, and 73.)

Abhishek Kumar, Bikash Sah, Arvind R Singh, Yan Deng, Xiangning He, Praveen Kumar, and Ramesh C Bansal. A review of multi criteria decision making (mcdm) towards sustainable renewable energy development. *Renewable and sustainable energy reviews*, 69:596–609, 2017. (Cited on page 1.)

F Kuo, I Sloan, Grzegorz Wasilkowski, and Henryk Woźniakowski. On decompositions of multivariate functions. *Mathematics of computation*, 79(270):953–966, 2010. (Cited on pages 143, 147, and 148.)

Christophe Labreuche. An axiomatization of the choquet integral in the context of multiple criteria decision making without any commensurability assumption. *Annals of Operations Research*, 271(2):701–735, 2018. (Cited on page 20.)

Christophe Labreuche and Michel Grabisch. Generalized Choquet-like aggregation functions for handling bipolar scales. *European Journal of Operational Research*, 172(3): 931–955, 2006a. (Cited on page 20.)

Christophe Labreuche and Michel Grabisch. Generalized choquet-like aggregation functions for handling bipolar scales. *European Journal of Operational Research*, 172(3): 931–955, 2006b. (Cited on page 21.)

Sébastien Lahaie. Kernel methods for revealed preference analysis. In *ECAI*, pages 439–444, 2010. (Cited on page 160.)

Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine learning research*, 5(Jan):27–72, 2004a. (Cited on pages 143, 150, 158, 159, 160, and 165.)

Gert RG Lanckriet, Tijl De Bie, Nello Cristianini, Michael I Jordan, and William Stafford Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004b. (Cited on page 158.)

Serge Lang. *Linear algebra*. Springer Science & Business Media, 1987. (Cited on page 143.)

John Langford, Lihong Li, and Tong Zhang. Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 10(3), 2009. (Cited on page 203.)

Julien Lesca and Patrice Perny. LP solvable models for multiagent fair allocation problems. In *ECAI 2010 Proceedings*, pages 393–398, 2010. (Cited on pages 20, 50, 63, and 209.)

Genyuan Li, Sheng-Wei Wang, Carey Rosenthal, and Herschel Rabitz. High dimensional model representations generated from low dimensional data samples. i. mp-cut-hdmr. *Journal of Mathematical Chemistry*, 30:1–30, 2001. (Cited on page 160.)

Yi Lin and Hao Helen Zhang. Component selection and smoothing in multivariate nonparametric regression. 2006. (Cited on page 161.)

Sijia Liu, Jie Chen, Pin-Yu Chen, and Alfred Hero. Zeroth-order online alternating direction method of multipliers: Convergence analysis and applications. In *International Conference on Artificial Intelligence and Statistics*, pages 288–297. PMLR, 2018. (Cited on pages 211 and 215.)

Sijia Liu, Parikshit Ram, Deepak Vijaykeerthy, Djallel Bouneffouf, Gregory Bramble, Horst Samulowitz, Dakuo Wang, Andrew Conn, and Alexander Gray. An admm based framework for automl pipeline configuration. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):4892–4899, Apr. 2020. doi: 10.1609/aaai.v34i04.5926. (Cited on page 212.)

Miguel Sousa Lobo, Lieven Vandenberghe, Stephen Boyd, and Hervé Lebret. Applications of second-order cone programming. *Linear algebra and its applications*, 284(1-3):193–228, 1998. (Cited on page 46.)

R Duncan Luce. Semiorders and a theory of utility discrimination. *Econometrica, Journal of the Econometric Society*, pages 178–191, 1956. (Cited on page 49.)

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017. (Cited on page 231.)

Mehrdad Mahdavi, Rong Jin, and Tianbao Yang. Trading regret for efficiency: online convex optimization with long term constraints. *The Journal of Machine Learning Research*, 13(1):2503–2528, 2012. (Cited on page 211.)

J-L Marichal. An axiomatic approach of the discrete choquet integral as a tool to aggregate interacting criteria. *IEEE transactions on fuzzy systems*, 8(6):800–807, 2000. (Cited on pages 19, 20, and 31.)

Hugo Martin and Patrice Perny. Incremental preference elicitation with bipolar choquet integrals. In *Algorithmic Decision Theory: 7th International Conference, ADT 2021, Toulouse, France, November 3–5, 2021, Proceedings 7*, pages 101–116. Springer, 2021. (Cited on page 21.)

Brendan McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 525–533. JMLR Workshop and Conference Proceedings, 2011. (Cited on pages 204 and 206.)

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. (Cited on page 230.)

James Mercer. Xvi. functions of positive and negative type, and their connection the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, 209(441-458): 415–446, 1909. (Cited on pages 150 and 155.)

Charles A Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(12), 2006. (Cited on page 153.)

Vincent Mousseau and Marc Pirlot. Preference elicitation and learning. *EURO Journal on Decision Processes*, 3(1):1–3, 2015. (Cited on page 26.)

Toshiaki Murofushi and Michio Sugeno. An interpretation of fuzzy measures and the choquet integral as an integral with respect to a fuzzy measure. *Fuzzy sets and Systems*, 29(2):201–227, 1989. (Cited on page 30.)

Arkadi S Nemirovski and Michael J Todd. Interior-point methods for optimization. *Acta Numerica*, 17:191–234, 2008. (Cited on page 47.)

Yurii Nesterov. A method for solving the convex programming problem with convergence rate o (1/k2). In *Soviet Mathematics Doklady*, volume 269, page 543, 1983. (Cited on page 44.)

Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259, 2009. (Cited on page 206.)

Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994. (Cited on page 46.)

Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018. (Cited on pages 36, 43, 44, 45, and 47.)

Robert Nishihara, Laurent Lessard, Ben Recht, Andrew Packard, and Michael Jordan. A general analysis of the convergence of admm. In *International conference on machine learning*, pages 343–352. PMLR, 2015. (Cited on page 212.)

Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980. (Cited on page 45.)

Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999. (Cited on pages 43 and 44.)

Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019. (Cited on pages 199, 201, 202, 203, 205, 206, and 218.)

Francesco Orabona, Luo Jie, and Barbara Caputo. Multi kernel learning with online-batch optimization. *Journal of Machine Learning Research*, 13(2), 2012. (Cited on page 174.)

Hua Ouyang, Niao He, Long Tran, and Alexander Gray. Stochastic alternating direction method of multipliers. In *International conference on machine learning*, pages 80–88. PMLR, 2013. (Cited on pages 211 and 215.)

Guillermo Owen. Multilinear extensions of games. *Management Science*, 18(5-part-2): 64–79, 1972. (Cited on page 21.)

Guillermo Owen. Multilinear extensions and the banzhaf value. *Naval research logistics quarterly*, 22(4):741–750, 1975. (Cited on page 107.)

Barbara Pękala, Anna Wilbik, Jarosław Szkoła, Krzysztof Dyczkowski, and Patryk Żywica. Federated learning with the choquet integral as aggregation method. In *2024 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE, 2024. (Cited on page 230.)

Guilherme D Pelegrina, Michel Grabisch, Leonardo T Duarte, and MT Romano. Multilinear model: New issues in capacity identification. In *From Multiple Criteria Decision Aid to Preference Learning (DA2PL'2018)*, 2018. (Cited on page 31.)

Guilherme Dean Pelegrina, Leonardo Tomazeli Duarte, Michel Grabisch, and João Marcos Travassos Romano. The multilinear model in multicriteria decision making: The case of 2-additive capacities and contributions to parameter identification. *European Journal of Operational Research*, 282(3):945–956, 2020a. (Cited on pages 2, 31, 52, and 106.)

Guilherme Dean Pelegrina, Leonardo Tomazeli Duarte, Michel Grabisch, and João Marcos Travassos Romano. The multilinear model in multicriteria decision making: The case of 2-additive capacities and contributions to parameter identification. *European Journal of Operational Research*, 282(3):945–956, 2020b. (Cited on page 198.)

Guilherme Dean Pelegrina, Leonardo Tomazeli Duarte, and Michel Grabisch. A k-additive choquet integral-based approach to approximate the shap values for local interpretability in machine learning. *Artificial Intelligence*, 325:104014, 2023. (Cited on page 231.)

Guilherme Dean Pelegrina, Patrick Kolpaczki, and Eyke Hüllermeier. Shapley value approximation based on k-additive games. *arXiv preprint arXiv:2502.04763*, 2025. (Cited on page 231.)

Patrice Perny. Modélisation des préférences, agrégation multicritere et systemes d'aide à la décision. *These d'habilitation, Université Pierre et Marie Curie*, 2000. (Cited on pages 7 and 10.)

Patrice Perny, Paolo Viappiani, and Abdellah Boukhatem. Incremental preference elicitation for decision making under risk with the rank-dependent utility model. In *proc. of UAI*, 2016. (Cited on page 64.)

Gabriella Pigozzi, Alexis Tsoukias, and Paolo Viappiani. Preferences in artificial intelligence. *Annals of Mathematics and Artificial Intelligence*, 77:361–401, 2016. (Cited on page 6.)

Anthony J. Pinar, Derek T. Anderson, Timothy C. Havens, Alina Zare, and Titilope Adeyeba. Measures of the Shapley index for learning lower complexity fuzzy integrals. *Granul. Comput.*, 2:303 – 319, 2017. (Cited on pages 3, 53, 58, and 106.)

Marc Pirlot and Philippe Vincke. *Semiorders: Properties, representations, applications*, volume 36. Springer Science & Business Media, 2013. (Cited on page 49.)

John C Platt. Fast training of support vector machines using sequential minimal optimization. 1998. (Cited on pages 47 and 158.)

Erik Pohl and Jutta Geldermann. Selection of multi-criteria energy efficiency and emission abatement portfolios in container terminals. *European Journal of Operational Research*, 316(1):386–395, 2024. (Cited on page 1.)

Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992. (Cited on page 203.)

Samira Pourkhajouei, Federico Toffano, Paolo Viappiani, and Nic Wilson. An efficient non-bayesian approach for interactive preference elicitation under noisy preference models. In *European Conference on Symbolic and Quantitative Approaches with Uncertainty*, pages 308–321. Springer, 2023. (Cited on pages viii, 193, 194, and 195.)

Michael JD Powell. A method for nonlinear constraints in minimization problems. *Optimization*, pages 283–298, 1969. (Cited on page 212.)

Henri Prade, Agnes Rico, and Mathieu Serrurier. Elicitation of sugeno integrals: A version space learning perspective. In *Foundations of Intelligent Systems: 18th International Symposium, ISMIS 2009, Prague, Czech Republic, September 14-17, 2009. Proceedings 18*, pages 392–401. Springer, 2009. (Cited on page 228.)

John W Pratt. Risk aversion in the small and in the large. In *Uncertainty in economics*, pages 59–79. Elsevier, 1978. (Cited on page 62.)

Zhiwei Tony Qin and Donald Goldfarb. Structured sparsity via alternating direction methods. *Journal of Machine Learning Research*, 13(5), 2012. (Cited on page 214.)

Shibin Qiu and Terran Lane. A framework for multiple kernel support vector regression and its applications to sirna efficacy prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(2):190–199, 2008. (Cited on pages 159, 160, and 165.)

John Quiggin. *Generalized expected utility theory: The rank-dependent model.* Springer Science, 2012. (Cited on pages 28, 59, 64, and 225.)

Herschel Rabitz and Ömer F Aliş. General foundations of high-dimensional model representations. *Journal of Mathematical Chemistry*, 25(2):197–233, 1999. (Cited on page 160.)

Filip Radlinski and Thorsten Joachims. Query chains: learning to rank from implicit feedback. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 239–248, 2005. (Cited on pages 120, 160, and 167.)

Alain Rakotomamonjy, Francis Bach, Stéphane Canu, and Yves Grandvalet. More efficiency in multiple kernel learning. In *Proceedings of the 24th international conference on Machinelearning*, pages 775–782, 2007. (Cited on pages 159, 165, and 174.)

Alain Rakotomamonjy, Francis Bach, Stéphane Canu, and Yves Grandvalet. Simplemkl. *Journal of Machine Learning Research*, 9:2491–2521, 2008. (Cited on pages 159, 160, 165, and 174.)

James O. Ramsay. Monotone regression spline in action. *Statistical Science*, page 425–441, 1988. (Cited on page 67.)

Saman Razavi, Anthony Jakeman, Andrea Saltelli, Clémentine Prieur, Bertrand Iooss, Emanuele Borgonovo, Elmar Plischke, Samuele Lo Piano, Takuya Iwanaga, William Becker, et al. The future of sensitivity analysis: an essential discipline for systems modeling and policy support. *Environmental Modelling & Software*, 137:104954, 2021. (Cited on page 145.)

James Renegar. A polynomial-time algorithm, based on newton's method, for linear programming. *Mathematical programming*, 40(1):59–93, 1988. (Cited on page 46.)

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951. (Cited on page 203.)

Guilherme V Rocha, Xing Wang, and Bin Yu. Asymptotic distribution and sparsistency for l1-penalized parametric m-estimators with applications to linear svm and logistic regression. *arXiv preprint arXiv:0908.1940*, 2009. (Cited on page 111.)

R Tyrrell Rockafellar. *Convex analysis*, volume 28. Princeton university press, 1997. (Cited on page 36.)

Volker Roth. The generalized lasso. *IEEE transactions on neural networks*, 15(1):16–28, 2004. (Cited on page 214.)

Volker Roth and Bernd Fischer. The group-lasso for generalized linear models: uniqueness of solutions and efficient algorithms. In *Proceedings of the 25th international conference on Machine learning*, pages 848–855, 2008. (Cited on page 42.)

Bernard Roy and Philippe Vincke. Multicriteria analysis: survey and new directions. *European journal of operational research*, 8(3):207–218, 1981. (Cited on pages 7 and 72.)

Bernard Roy et al. Méthodologie multicritère d'aide à la décision. 1985. (Cited on page 7.)

Norbert Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145–147, 1972. (Cited on page 195.)

Craig Saunders, Alexander Gammerman, and Volodya Vovk. Ridge regression learning algorithm in dual variables. 1998. (Cited on page 162.)

Alistair Savage. Linear algebra i. 2018. (Cited on page 143.)

Leonard J Savage. *The foundations of statistics*. John Wiley & Sons, 1954. (Cited on pages 7 and 61.)

Isabelle Sayn. Recourir à un barème pour fixer la prestation compensatoire? portées et limites de l'outil. *Le traitement juridique des conséquences économiques du divorce, Une approche économique, sociologique et juridique de la prestation compensatoire*, page 151, 2018. (Cited on page 135.)

PB Schiilkop, Chris Burgest, and Vladimir Vapnik. Extracting support data for a given task. In *Proceedings, First International Conference on Knowledge Discovery & Data Mining. AAAI Press, Menlo Park, CA*, pages 252–257, 1995. (Cited on page 118.)

David Schmeidler. Subjective probability and expected utility without additivity. *Econometrica*, 57(3):571–587, 1989. (Cited on pages 1, 14, 20, and 62.)

B Schölkopf. Learning with kernels: support vector machines, regularization, optimization, and beyond, 2002. (Cited on pages 119, 143, 150, 154, 155, 157, and 161.)

Nicol N Schraudolph, Jin Yu, and Simon Günter. A stochastic quasi-newton method for online convex optimization. In *Artificial intelligence and statistics*, pages 436–443. PMLR, 2007. (Cited on page 45.)

Lesia Semenova, Cynthia Rudin, and Ronald Parr. On the existence of simpler machine learning models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1827–1858, 2022. (Cited on page 71.)

G Shafer. *A Mathematical Theory of Evidence.* Princeton University Press, 1976. (Cited on page 78.)

Shai Shalev-Shwartz. *Online learning: Theory, algorithms, and applications.* PhD thesis, Hebrew University, 2007. (Cited on pages 44 and 204.)

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012. (Cited on pages 44, 198, 199, 200, 203, 204, and 205.)

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms.* Cambridge university press, 2014. (Cited on page 33.)

Lloyd S Shapley. Cores of convex games. *International journal of game theory*, 1:11–26, 1971. (Cited on page 17.)

John Shawe-Taylor, Nello Cristianini, et al. *Kernel methods for pattern analysis.* Cambridge university press, 2004. (Cited on pages 119, 122, 124, and 150.)

Ivan Singer. Extensions of functions of 0-1 variables and applications to combinatorial optimization. *Numerical Functional Analysis and Optimization*, 7(1):23–62, 1985. (Cited on page 22.)

Eleftherios Siskos and Peter Burgherr. Multicriteria decision support for the evaluation of electricity supply resilience: Exploration of interacting criteria. *European Journal of Operational Research*, 298(2):611–626, 2022. (Cited on page 1.)

Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14:199–222, 2004. (Cited on page 155.)

Ilya M Sobol'. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and computers in simulation*, 55(1-3):271–280, 2001. (Cited on pages 143, 145, and 161.)

Ilya M Sobol'. Theorems and examples on high dimensional model representation. *Reliability Engineering and System Safety*, 79(2):187–193, 2003. (Cited on page 148.)

Nataliya Sokolovska, Yann Chevaleyre, Karine Clément, and Jean-Daniel Zucker. The fused lasso penalty for learning interpretable medical scoring systems. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 4504–4511. IEEE, 2017. (Cited on page 229.)

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18990–18998, 2024. (Cited on page 6.)

Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *The Journal of Machine Learning Research*, 7:1531–1565, 2006. (Cited on pages 160 and 174.)

Suvrit Sra, Sebastian Nowozin, and Stephen J Wright. *Optimization for machine learning*. MIT press, 2011. (Cited on page 43.)

Ingo Steinwart. *Support Vector Machines*. Springer, 2008. (Cited on pages 150 and 155.)

Ralph E Steuer. Multiple criteria optimization. *Theory, computation, and application*, 1986. (Cited on page 29.)

Mark Stitson, Alex Gammerman, Vladimir Vapnik, Volodya Vovk, Chris Watkins, and Jason Weston. Support vector regression with anova decomposition kernels. *Advances in kernel methods—Support vector learning*, pages 285–292, 1999. (Cited on page 162.)

Michio Sugeno. Theory of fuzzy integrals and its applications. *Doctoral Thesis, Tokyo Institute of Technology*, 1974. (Cited on pages 1, 15, and 24.)

Michio Sugeno. *Fuzzy measures and fuzzy integrals: A survey*, page 89–102. North-Holland, Amsterdam, 1977. (Cited on pages 223 and 228.)

Peiqi Sun, Michel Grabisch, and Christophe Labreuche. An improvement of random node generator for the uniform generation of capacities. *Annals of Mathematics and Artificial Intelligence*, pages 1–26, 2023. (Cited on page 90.)

Shiliang Sun, Zehui Cao, Han Zhu, and Jing Zhao. A survey of optimization methods from a machine learning perspective. *IEEE transactions on cybernetics*, 50(8):3668–3681, 2019. (Cited on page 45.)

Taiji Suzuki. Dual averaging and proximal gradient descent for online alternating direction multiplier method. In *International Conference on Machine Learning*, pages 392–400. PMLR, 2013. (Cited on pages 211, 214, 215, and 217.)

Taiji Suzuki. Stochastic dual coordinate ascent with alternating direction method of multipliers. In *International Conference on Machine Learning*, pages 736–744. PMLR, 2014. (Cited on page 211.)

Ali Fallah Tehrani. The Choquet kernel on the use of regression problem. *Information Sciences*, 556:256–272, 2021. (Cited on pages 3, 52, 106, 120, 127, and 160.)

Ali Fallah Tehrani and Diane Ahrens. Modeling label dependence for multi-label classification using the choquistic regression. *Pattern Recognition Letters*, 92:75–80, 2017. (Cited on page 86.)

Ali Fallah Tehrani and Eyke Hüllermeier. Ordinal Choquistic regression. In *EUSFLAT*, 2013. (Cited on page 198.)

Ali Fallah Tehrani and Eyke Hüllermeier. Ordinal Choquistic regression. In *8th conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-13)*, pages 842–849, 2013. (Cited on pages 52, 59, 89, and 106.)

Ali Fallah Tehrani, Weiwei Cheng, Krzysztof Dembczyński, and Eyke Hüllermeier. Learning monotone nonlinear models using the Choquet integral. *Machine Learning*, 89:183 – 211, 2012a. (Cited on pages 52 and 59.)

Ali Fallah Tehrani, Weiwei Cheng, Krzysztof Dembczynski, and Eyke Hüllermeier. Learning monotone nonlinear models using the Choquet integral. *Machine Learning*, 89(1-2): 183–211, 2012b. (Cited on page 3.)

Ali Fallah Tehrani, Weiwei Cheng, and Eyke Hülermeier. Preference learning using the Choquet integral: the case of multipartite ranking. *IEEE Transactions on Fuzzy Systems*, 20(6):1102 – 1113, 2012c. (Cited on pages 51 and 52.)

Ali Fallah Tehrani, Christophe Labreuche, and Eyke Hüllermeier. Choquistic utilitaristic regression. In *DA2PL*, pages 35 – 42, 2014a. (Cited on pages 3, 52, and 59.)

Ali Fallah Tehrani, Marc Strickert, and Eyke Hüllermeier. The choquet kernel for monotone data. In *Esann*, 2014b. (Cited on pages 120, 123, 124, and 160.)

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (methodological)*, 58(1):267 – 88, 1996. (Cited on pages 2, 38, and 40.)

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(1):91–108, 2005. (Cited on pages 42 and 229.)

Mikhail Timonin. Axiomatization of the choquet integral for 2-dimensional heterogeneous product sets. *arXiv preprint arXiv:1507.04167*, 2015. (Cited on page 20.)

Simon Tong and Daphne Koller. Support vector machine active learning with application sto text classification. In *Proceedings of the seventeenth international conference on machine learning*, pages 999–1006, 2000. (Cited on page 181.)

Vicenç Torra. The weighted owa operator. *International Journal of Intelligent Systems*, 12(2):153–166, 1997. (Cited on page 1.)

Amos Tversky. On the elicitation of preferences: Descriptive and prescriptive considerations. *Conflicting objectives in decisions*, pages 209–222, 1977. (Cited on page 26.)

Amos Tversky and Daniel Kahneman. An analysis of decision under risk. *Econometrica*, 47(2):263–292, 1979. (Cited on page 20.)

Amos Tversky and Daniel Kahneman. The framing of decisions and the psychology of choice. *science*, 211(4481):453–458, 1981. (Cited on pages 1 and 6.)

Sara van de Geer. ℓ1-regularization in high-dimensional statistical models. In *Proceedings of the International Congress of Mathematicians 2010 (ICM 2010)*, pages 2351–2369, 2010. (Cited on page 87.)

Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM review*, 38 (1):49–95, 1996. (Cited on page 46.)

Daniel Vanderpooten and Philippe Vincke. Description and analysis of some representative interactive multicriteria procedures. In *Models and Methods in Multiple Criteria Decision Making*, pages 1221–1238. Elsevier, 1989. (Cited on page 29.)

Vladimir Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991. (Cited on page 34.)

Vladimir Vapnik. Statistical learning theory. *John Wiley & Sons google schola*, 2:831–842, 1998. (Cited on page 162.)

Vladimir Vapnik, Steven Golowich, and Alex Smola. Support vector method for function approximation, regression estimation and signal processing. *Advances in neural information processing systems*, 9, 1996. (Cited on page 163.)

Vladimir N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, 1995. ISBN 978-0387987804. doi: 10.1007/978-1-4757-2440-0. (Cited on pages 2, 33, 35, 150, 184, 196, and 230.)

VN Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264, 1971. (Cited on page 34.)

Manik Varma and Debajyoti Ray. Learning the discriminative power-invariance trade-off. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007. (Cited on pages 158, 159, 160, 163, and 174.)

J Von Neumann and O Morgenstern. Theory of games and economic behavior. 1944. (Cited on pages 1, 6, 8, and 62.)

Detlof Von Winterfeldt and Ward Edwards. *Decision analysis and behavioral research*. Cambridge University Press, 1986. (Cited on pages 27 and 64.)

John von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1947. (Cited on page 1.)

Willem Waegeman, Bernard De Baets, and Luc Boullart. Kernel-based learning methods for preference aggregation. *4OR*, 7(2):169–189, 2009. (Cited on pages 120, 123, and 160.)

Grace Wahba. *Spline models for observational data*. SIAM, 1990. (Cited on pages 161 and 162.)

Grace Wahba, Yuedong Wang, Chong Gu, Ronald Klein, and Barbara Klein. Smoothing spline anova for exponential families, with application to the wisconsin epidemiological study of diabetic retinopathy: the 1994 neyman memorial lecture. *The Annals of Statistics*, 23(6):1865–1895, 1995. (Cited on page 161.)

Peter Wakker and Daniel Deneffe. Eliciting von neumann-morgenstern utilities when probabilities are distorted or unknown. *Management science*, 42(8):1131–1150, 1996. (Cited on pages 28, 59, and 64.)

Peter P Wakker. Testing and characterizing properties of nonadditive measures through violations of the sure-thing principle. *Econometrica*, 69(4):1039–1059, 2001. (Cited on page 62.)

Hansheng Wang, Guodong Li, and Guohua Jiang. Robust regression shrinkage and consistent variable selection through the LAD-lasso. *Journal of Business & Economic Statistics*, 25(3):347–355, 2007. (Cited on page 82.)

Huahua Wang and Arindam Banerjee. Online alternating direction method. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1699–1706, 2012. (Cited on pages 211, 213, 215, 217, and 218.)

Junxiang Wang, Fuxun Yu, Xiang Chen, and Liang Zhao. Admm for efficient deep learning with global convergence. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 111–119, 2019. (Cited on page 212.)

Tianhan Wang and Craig Boutilier. Incremental utility elicitation with the minimax regret decision criterion. In *Ijcai*, volume 3, pages 309–316, 2003. (Cited on pages 30, 176, 177, and 178.)

Manfred K Warmuth, Arun K Jagota, et al. Continuous and discrete-time nonlinear gradient descent: Relative loss bounds and convergence. In *Electronic proceedings of the 5th International Symposium on Artificial Intelligence and Mathematics*, volume 326. Citeseer, 1997. (Cited on page 203.)

Chelsea C White, Andrew P Sage, and Shigeru Dozono. A model of multiattribute decisionmaking and trade-off weight determination under uncertainty. *IEEE Transactions on Systems, Man, and Cybernetics*, (2):223–229, 1984. (Cited on pages 30, 176, and 177.)

Andrzej P Wierzbicki. On the completeness and constructiveness of parametric characterizations to vector optimization problems. *Operations-Research-Spektrum*, 8(2):73–87, 1986. (Cited on pages 1 and 23.)

Andrzej P Wierzbicki. A methodological guide to multiobjective optimization. In *Optimization Techniques: Proceedings of the 9th IFIP Conference on Optimization Techniques Warsaw, September 4–8, 1979*, pages 99–123. Springer, 2005. (Cited on page 29.)

Christopher M Wilson, Kaiqiao Li, Xiaoqing Yu, Pei-Fen Kuan, and Xuefeng Wang. Multiple-kernel learning for genomic data mining and prediction. *BMC bioinformatics*, 20:1–7, 2019. (Cited on page 158.)

Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017. (Cited on pages 3 and 32.)

Tong Tong Wu and Kenneth Lange. Coordinate descent algorithms for lasso penalized regression. 2008. (Cited on page 46.)

Xiaofei Wu, Rongmei Liang, and Hu Yang. Penalized and constrained LAD estimation in fixed and high dimension. *Statistical Papers*, pages 1–43, 2022. (Cited on page 87.)

Lin Xiao. Dual averaging method for regularized stochastic learning and online optimization. *Advances in Neural Information Processing Systems*, 22, 2009. (Cited on pages 44, 204, and 206.)

Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, 2010. (Cited on pages 3, 198, 204, 205, 206, and 218.)

Jinfeng Xu and Zhiliang Ying. Simultaneous estimation and variable selection in median regression using lasso-type penalty. *Annals of the Institute of Statistical Mathematics*, 62:487–514, 2010. (Cited on page 87.)

Zenglin Xu, Rong Jin, Shenghuo Zhu, Michael Lyu, and Irwin King. Smooth optimization for effective multiple kernel learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pages 637–642, 2010. (Cited on page 159.)

Menahem E Yaari. The Dual Theory of Choice under Risk. *Econometrica*, 55(1):95–115, 1987. (Cited on page 64.)

Ronald R Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):183–190, 1988. (Cited on page 1.)

Guo-Xian Yu, Huzefa Rangwala, Carlotta Domeniconi, Guoji Zhang, and Zili Zhang. Protein function prediction by integrating multiple kernels. In *IJCAI*, pages 1869–1875, 2013. (Cited on page 158.)

Hao Yu and Michael J Neely. A low complexity algorithm with o (t) regret and o (1) constraint violations for online convex optimization with long term constraints. *Journal of Machine Learning Research*, 21(1):1–24, 2020. (Cited on page 211.)

Xueying Zhan, Huan Liu, Qing Li, and Antoni B Chan. A comparative survey: Benchmarking for pool-based active learning. In *IJCAI*, pages 4679–4686, 2021. (Cited on page 181.)

Chicheng Zhang. Efficient active learning of sparse halfspaces. In *Conference on Learning Theory*, pages 1856–1880. PMLR, 2018. (Cited on page 223.)

Chicheng Zhang, Jie Shen, and Pranjal Awasthi. Efficient active learning of sparse halfspaces with arbitrary bounded noise. *Advances in Neural Information Processing Systems*, 33:7184–7197, 2020. (Cited on page 223.)

Xiaoqun Zhang, Martin Burger, and Stanley Osher. A unified primal-dual algorithm framework based on bregman iteration. *Journal of Scientific Computing*, 46:20–46, 2011. (Cited on page 217.)

Peng Zhao and Bin Yu. On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7:2541–2563, 2006. (Cited on pages 83 and 84.)

Zhou Zhao, Hanqing Lu, Deng Cai, Xiaofei He, and Yueting Zhuang. User preference learning for online social recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 28(9):2522–2534, 2016. (Cited on page 198.)

Qi Zheng, Colin Gallagher, and KB Kulasekera. Robust adaptive lasso for variable selection. *Communications in Statistics-Theory and Methods*, 46(9):4642–4659, 2017. (Cited on page 87.)

Alexander Zien and Cheng Soon Ong. Multiclass multiple kernel learning. In *Proceedings of the 24th international conference on Machine learning*, pages 1191–1198, 2007. (Cited on page 159.)

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003. (Cited on pages 201 and 202.)

Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006. (Cited on pages 81, 84, 86, 87, and 126.)

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B (methodological)*, 67(2):301 – 320, 2005. (Cited on pages 42 and 89.)

Abstract. The work presented in this thesis lies at the intersection of decision theory and machine learning. The objective is to develop learning methods for preference models grounded in decision theory, to explain or predict a decision maker's preferences and ultimately recommend optimal alternatives in decision problems. We focus in particular on value function models that account for interactions between different viewpoints on the alternatives, such as the Choquet integral, the multilinear utility, and decomposable GAI utility functions. These models possess strong descriptive power, while also ensuring a form of rationality in preferences through the satisfaction of desirable mathematical properties, and allowing for interpretability via their parameters. Due to the combinatorial nature of interactions, learning such models poses a computational challenge, as it requires determining an exponential number of parameters, sometimes subject to combinatorial constraints. In this thesis, we propose to control the flexibility of these models through the learning of sparse representations of interactions, notably through the use of sparsity-inducing regularizations, and to reduce computational complexity by leveraging convex optimization methods from machine learning suited to high-dimensional sparse learning problems. In summary, this thesis contributes by (i) providing learning problem formulations tailored to various preference models and learning settings: from pre-collected examples (passive learning), from carefully selected queries (preference elicitation or active learning), or from a stream of examples (online learning), (ii) developing computationally efficient optimization algorithms to solve these problems, and (iii) conducting experimental evaluations on both synthetic and real-world preference data.

Keywords: decision theory, preference elicitation, multicriteria decision-making, decision-making under uncertainty, aggregation function, machine learning, preference learning, convex optimization.

Résumé. Les travaux présentés dans cette thèse se situent à l'intersection de la théorie de la décision et de l'apprentissage automatique. L'objectif est de proposer des méthodes d'apprentissage pour des modèles de préférences issus de la théorie de la décision, dans le but d'expliquer ou de prédire les préférences d'un décideur, et, en fin de compte, de recommander des alternatives optimales dans des problèmes de décision. Nous nous intéressons en particulier aux modèles de type fonction de valeur prenant en compte les interactions entre les différents points de vue sur les alternatives, tels que l'intégrale de Choquet, l'utilité multilinéaire et les fonctions d'utilité GAI décomposables. Ces modèles présentent un fort pouvoir descriptif, tout en assurant une forme de rationalité dans les préférences via le respect de propriétés mathématiques souhaitables, et en permettant l'interprétabilité grâce à leurs paramètres. En raison de la nature combinatoire des interactions, l'apprentissage de ces modèles représente un défi computationnel, car il nécessite la détermination d'un nombre exponentiel de paramètres, parfois soumis à des contraintes combinatoires. Dans cette thèse, nous proposons de contrôler la flexibilité de ces modèles via l'apprentissage de représentations parcimonieuses des interactions, notamment à l'aide de régularisations favorisant la parcimonie, et d'alléger la complexité computationnelle en exploitant des méthodes d'optimisation convexes pour l'apprentissage automatique, adaptées à l'apprentissage parcimonieux en grande dimension. En résumé, cette thèse fournit (i) des formulation de problèmes d'apprentissage adaptées à différents modèles de préférences ainsi qu'à divers cadres d'apprentissage : à partir d'exemples préalablement collectés (apprentissage passif), à partir de requêtes soigneusement sélectionnées (élicitation des préférences ou apprentissage actif), ou à partir de flux d'exemples (apprentissage en ligne), (ii) des algorithmes d'optimisation computationnellement efficaces pour la résolution de ces problèmes, et (iii) des évaluations expérimentales sur des données de préférences synthétiques et réelles.

Mots-clés : théorie de la décision, élicitation des préférences, décision multicritère, décision dans l'incertains, fonction d'agrégation, apprentissage des préférences, optimisation convexe.