# Holistic Preference Learning with the Choquet Integral

**Bénédicte GOUJON Christophe LABREUCHE**

Thales Research & Technology France

**Abstract**

The current approaches to construct a multi-criteria model based on a Choquet integral are split into two separate steps: construct first the utility functions and then the aggregation function. Unfortunately, the decision maker may feel some difficulties in addressing these tricky steps. In this paper, we propose a preference learning algorithm that constructs both the utility functions and the capacity from several preferences or evaluations. The algorithm is based on a fixed-point approach that transforms the global optimization learning problem into two iterative linear problems. Each problem objective is to minimize the number of non-validated learning examples.

**Keywords**: Preference Learning, Multi-criteria Decision Model, Choquet Integral, Fixed-point.

## 1. Introduction

Multi-Criteria Decision Making (MCDM) consists in helping a decision maker (DM) in choosing one option among a set of candidate alternatives, on the basis of several and conflicting criteria. Many models have been developed to support a DM [1]. A family of models called *decomposable* [2] takes the form of an aggregation function applied on partial utility functions applied on the various attributes. Among the variety of existing aggregation functions, the weighted sum is one of the most classical and widely used. It has the major drawback of assuming independence among criteria. In order to overcome this difficulty, the Choquet integral has been introduced in MCDM [3]. It has the ability to represent interaction among criteria ranging from veto, favor to complementarity or redundancy between criteria [4].

From an application side, it is very important to identify the parameters of the Choquet integral – namely the capacity (also called fuzzy measure). The traditional approach to elicit a MCDM model based on the Choquet integral consists in two separate steps [5]: first, construct the utility function for each attribute, and then construct the capacity once the utility functions are known. A lot of papers have focused on the second phase, transforming it into an optimization problem, once the DM has provided a set of learning examples [6]. The first stage (construction of the utility functions) follows the MACBETH approach [7]. It assumes that the DM is able to identify on each attribute two values – called reference levels – that have a special meaning (the first one is neutral while the second one is satisfycing). This assumption appears to be too drastic in some applications.

The idea is then to learn at the same time both the utility functions and the capacity. The importance of doing so has been emphasized in [8]. Very few theoretical works can be found on MCDM models composed of both the Choquet integral and its utility functions [9, 10]. The determination of not only the admissible capacities but also the utility functions has been considered from a practical side only in one paper [11]. The existing methods to solve this problem as a whole are based on stochastic approaches (Monte Carlo or genetic algorithm), as it is the case in [11].

The problem of learning a preference model has recently become a hot topic in Artificial Intelligence under the name of *preference learning* [12]. Preference learning can be used for solving different tasks, among which one can find *object ranking* or *learning to rank*. Although there are some similarities between preference learning and the aim of this paper, there are some clear distinctions. In our approach, we have only a few learning examples so that we impose that each of them is satisfied (constraint approach). In preference learning, the set of learning data is much larger and the determination of the preference model is often turned into a regression problem (when a numerical score is constructed). The learning data are not considered as hard constraints in this approach.

In this paper, we propose a preference learning algorithm to replace the two previously mentioned steps by automating the calculation of the utility functions and the capacity, using preferences or evaluations expressed by the DM on various solutions. For example, when the decision consists in buying a home, the expert has just to express some preferences between some houses, and not to describe in details the utility function on each attribute and each attribute importance. Such examples are called holistic. The determination of both the utility functions and the capacity from these holistic learning examples can be translated into an optimization problem. Unfortunately, this latter is very complex as it is not linear and the constraints are even not convex. We propose an approximate

algorithm based on a fixed-point approach. The main idea is to transform the original optimization problem into two iterative linear problems. The aim of the algorithm is not to find the most appropriate values of utility functions and capacity. It only tries to check whether there exist values of the parameters which fulfil the learning examples.

The layout of the paper is as follows. Section 2 presents the multi-criteria decision model that is based on the two-additive Choquet integral. Then in Section 3, we consider the inputs and issues of the problem. The algorithm is detailed in Section 4 and illustrated with an example (Section 5).

## 2. Multi-criteria decision model

We are given a set of $n$ attributes indexed by $N = \{1, \ldots, n\}$. Each attribute $i \in N$ is represented by a set $X_i$ which can be discrete or continuous (an interval). The alternatives are characterized by a value on each attribute, and are thus represented by an element of $X = X_1 \times \cdots \times X_n$. We aim to represent the overall assessment of a decision maker

$$H : X \to \mathbb{R}.$$

The most commonly form for $H$ is the *decomposable* model [13] and take the form $H(x) = F(u_1(x_1), \ldots, u_n(x_n))$, where the $u_i$'s : $X_i \to [0, 1]$ are called the *utility functions* (also called *value functions*) and $F : [0, 1]^n \to [0, 1]$ is an *aggregation function*. Examples of aggregation functions are the weighted sum or the Choquet integral [14, 3, 4]. We will consider in this paper a Choquet integral.

We assume that we are given a preference relation $\succsim_i$ over each attribute $i$. Utility functions $u_i$ are assumed to fulfil the following monotonicity conditions:

$$\forall x_i, y_i \in X_i \quad y_i \succsim_i x_i \;\Rightarrow\; u_i(y_i) \geq u_i(x_i) \tag{1}$$

### 2.1. Background on the Choquet integral

The Choquet integral is based on a capacity.

**Definition 1** *A* fuzzy measure *[15] or* capacity *[14] on $N$ is a set function $\mu : 2^N \to [0, 1]$ satisfying*

- *Monotonicity: $A \subseteq B \Rightarrow \mu(A) \leq \mu(B)$,*
- *Normalization: $\mu(\emptyset) = 0$, $\mu(N) = 1$.*

The Möbius transform (see e.g. [16]) of $\mu$ is defined by

$$m(A) = \sum_{B \subseteq A} (-1)^{|A \setminus B|} \mu(B). \tag{2}$$

Reciprocally, $\mu$ can be recovered from the Möbius transform by

$$\mu(A) = \sum_{B \subseteq A} m(B). \tag{3}$$

Note that the monotonicity and normalization conditions on $\mu$ can easily be translated in terms of conditions on $m$:

$$\forall i \in N \; \forall A \subseteq N \setminus \{i\} \quad \sum_{B \subseteq A} m(B \cup \{i\}) \geq 0 \tag{4}$$

$$\sum_{A \subseteq N} m(A) = 1 \tag{5}$$

The Choquet integral of $a \in [0, 1]^N$ w.r.t. the Möbius coefficients $m$ (also called the *Lovász extension*) is defined by [14]

$$C_m(a) = \sum_{A \subseteq N} m(A) \cdot \bigwedge_{i \in A} a_i \;, \quad \forall a \in [0, 1]^N \tag{6}$$

where $m$ is the Möbius transform of $\mu$, and $\wedge$ is the min operator.

### 2.2. Two-additive Choquet integral

We first start with the concept of $k$ additive capacity, which is a particular family of capacities.

**Definition 2** *[17] Let $k \in \{1, \ldots, n-1\}$. A capacity $\mu$ is said to be $k$-additive if $m(A) = 0$ whenever $|A| > k$, and there exists some $A \subseteq N$ with $|A| = k$ such that $m(A) \neq 0$.*

An important particular case of the Choquet integral is the 2-additive case. We use the following notation

$$m_i = m(\{i\}) \quad \text{and} \quad m_{i,j} = m(\{i, j\}).$$

The monotonicity and normalization conditions become in this case

$$\forall i \in N \; \forall A \subseteq N \setminus \{i\}$$

$$m_i + \sum_{j \in A} m_{i,j} \geq 0 \tag{7}$$

$$\sum_{i \in N} m_i + \sum_{\{i,j\} \subseteq N} m_{i,j} = 1 \tag{8}$$

The Choquet integral of $a \in [0, 1]^N$ w.r.t. the 2-additive Möbius coefficients $m$ is

$$C_m(a) = \sum_{i \in N} m_i \, a_i + \sum_{\{i,j\} \subseteq N} m_{i,j} \, a_i \wedge a_j. \tag{9}$$

### 2.3. Commensurability

The overall utility $H$ takes the form

$$H(x) = C_\mu(u_1(x_1), \ldots, u_n(x_n)). \tag{10}$$

We have seen that the expression of the Choquet integral uses minimum functions of its integrand. This means that the values $u_i(x_i)$ and $u_j(x_j)$ shall be comparable. Two scales $u_i, u_j$ over criteria $i$ and $j$ are said to be *commensurate* if for every $x_i, x_j$ such that $u_i(x_i) = u_j(x_j)$, the degrees of satisfaction felt by the DM on criteria $i$ and $j$ are equal. Hence

utility functions $u_i$ shall transforming the attributes into a commensurate scale (representing some satisfaction degree).

We say that several values on different attributes are made commensurable, if one is able to compare them through a binary relation $\trianglerighteq$. For $x_i \in X_i$ and all $x_j \in X_j$, $\trianglerighteq$ is defined by

$$x_i \trianglerighteq x_j \quad \Longleftrightarrow \quad u_i(x_i) \geq u_j(x_j). \tag{11}$$

A preamble to the determination of all utility functions is to be able to construct $\trianglerighteq$.

## 3. Inputs and description of the problem to solve

This section presents the learning data that are used as well as a representation of the learning problem as an optimization problem.

### 3.1. Inputs: preferential information

The model input information consists in a set of relevant attributes related to a multi-criteria decision problem. On each attribute, we assume we are given the partial preference relation $\succsim_i$. In particular, we know whether the utility is increasing or decreasing. The input information used as learning data is composed of preferences between alternatives or evaluated alternatives:

- When an alternative is evaluated, the input is composed of a pair $(x, V_x)$ where $x \in X$ is an alternative and $V_x \in [0,1]$ is the corresponding evaluation. This preferential information is translated into the following relation:

$$H(x) = V_x. \tag{12}$$

- When a preference is expressed between two alternatives, the input is composed of a pair $(x, y)$, where $x, y \in X$ are two alternatives and $x$ is supposed to be less preferred to $y$. This preferential information is translated into the following relation:

$$H(x) \leq H(y). \tag{13}$$

To sum-up, the set of preferential information denoted by $\mathcal{P}$, is composed of pairs of the form $(x, V_x)$ and $(x, y)$.

### 3.2. Translation of the learning data into an optimization problem

According to (9), the global satisfaction calculation is based on the following equation, for every $x \in X$:

$$H(x) = \sum_{i \in N} m_i u_i(x_i) + \sum_{\{i,j\} \subseteq N} m_{i,j} u_i(x_i) \wedge u_j(x_j). \tag{14}$$

Let

$$\widehat{X} := \{x \in X \ : \ (x, V_x) \in \mathcal{P} \text{ or } (x, y) \in \mathcal{P} \text{ or } (y, x) \in \mathcal{P}\}.$$

the set of options that used in the learning examples. Moreover, we set for every $i \in N$

$$\widehat{X}_i = \{x_i \ : \ x \in \widehat{X}\}$$

the set of values on attribute $i$ used in the learning examples. Then the unknowns concerning the utility on attribute $i$ are $u_i(x_i)$ for every $x_i \in \widehat{X}_i$:

$$\widehat{u}_i := \{u_i(x_i) \ : \ x_i \in \widehat{X}_i\}$$

and

$$\widehat{u} := (\widehat{u}_1, \ldots, \widehat{u}_n).$$

The set of vectors $\widehat{u}$ is denoted by $\widehat{\mathcal{U}}$. We can translate the monotonicity conditions (1) in the following way

$$\forall x_i, y_i \in \widehat{X}_i \quad y_i \succsim_i x_i \ \Rightarrow \ u_i(y_i) \geq u_i(x_i) \tag{15}$$

The Möbius coefficients are also unknowns:

$$\widehat{m} := \{m_i \ : \ i \in N\} \cup \{m_{i,j} \ : \ \{i,j\} \subseteq N\}.$$

The set of vectors $\widehat{m}$ is denoted by $\widehat{\mathcal{M}}$. In total, the unknowns are basically pairs of the form

$$(\widehat{u}, \widehat{m}) \in \widehat{\mathcal{U}} \times \widehat{\mathcal{M}}.$$

Our aim is to find $(\widehat{u}, \widehat{m})$ that fulfils the monotonicity and normalization conditions (7), (8) and (15), and the conditions (12) and (13) for all learning examples in $\mathcal{P}$.

The major difficulty of this statement is that the constraints (12) and (13) are not linear in the unknowns $(\widehat{u}, \widehat{m})$. More precisely, they are bilinear. Hence Linear Programming cannot be used to determine $(\widehat{u}, \widehat{m})$ directly. Compare to the case when the utility functions $\widehat{u}$ are already set, the problem we address is much more complex.

## 4. Preference learning algorithm related to the Choquet integral

The idea of our algorithm is to use a fixed-point approach. Starting from an initial $\widehat{u}$, we want to find $\widehat{m}$ that fulfils (7), (8) and and the conditions (12) and (13) for all learning examples in $\mathcal{P}$. This problem is linear. Given $\widehat{u}$, there might be no solution to the previous constraint satisfaction problem. In this case, we look for the solution that violates as few learning examples as possible. This provides a value for $\widehat{m}$. Given this latter value, one then tries to find a new value for $\widehat{u}$ that fulfils (15), and the conditions (12) and (13) for all learning examples in $\mathcal{P}$. When this problem has no solution, we seek

for the solution which violates as few learning examples as possible. This provides a new value of $\widehat{u}$. And so on. The algorithm stops when one succeeds to satisfy all learning examples or when a maximal number of iterations is reached.

If the previous fixed-point algorithms fails to converge to a solution that satisfies to all learning examples, then the idea is to execute the same algorithm starting from another value of $\widehat{u}$.

First, we describe how to generate an initial utility vector $\widehat{u}$.

### 4.1. Generation of an initial utility order

To initialize the algorithm, we have to provide a first set of utility values. To do so, we first generate a possible global order on utilities, then we associate ordered values between 0 and 1 to each utility value.

On each attribute $k$, $n_k$ denotes the number of distinct values. The elements of $\widehat{X}_k$ are denoted by $x_{k,1}, x_{k,2}, \ldots, x_{k,n_k}$ with

$$x_{k,1} \precsim_i x_{k,2} \precsim_i \cdots \precsim_i x_{k,n_k}. \tag{16}$$

Let $u_{k,i} = u_k(x_{k,i})$. As discussed in Section 2.3, in order to be able to construct $u_{k,i}$ for all $i$ and $k$, one needs first to make all elements $\{x_{k,i} \ : \ k \in N \ , \ i \in \{1, \ldots, n_k\}\}$ commensurate and thus to define a preference relation $\trianglelefteq$ among these elements.

From (16), we have for each attribute $k$

$$u_{k,1} \leq u_{k,2} \leq \ldots \leq u_{k,n_k}.$$

We note this order by the following sequence

$$R_{\widehat{u}_k} = (u_{k,1}, u_{k,2}, \ldots, u_{k,n_k}).$$

From those utility orders for each attribute, we have to generate a relevant global order $\trianglelefteq$ on all utilities, denoted $R_{\widehat{u}}$.

Algorithm 1 describes the generation of a global order on utilities on all attributes. It starts with an empty $R_{\widehat{u}}$. At the beginning one can select any of the element of $u_{1,1}, u_{2,1}, u_{3,1}, \ldots, u_{n,1}$ (the least preferred values on each attribute). In order to choose such element, one only need to select $i \in N$. Set *next* $\subseteq N$ in Algorithm 1 below is the set of criteria from which one can still select one element. We start with *next* $= N$. If we have chosen $u_{2,1}$ at the first iteration, one needs to select the second element among $u_{1,1}, u_{2,2}, u_{3,1}, \ldots, u_{n,1}$. We set $h_i = 1$ for all $i \in N$, at the beginning of the algorithm. If term $u_{2,1}$ is chosen, then we update $h_2 = 2$. Hence if $i \in N$ is selected at the current iteration, we add $u_{i,h_i}$ to $R_{\widehat{u}}$, and $h_i$ is replaced by $h_i + 1$ unless $h_i = n_i$ in which case $i$ is removed from *next* ($i$ cannot be selected anymore as there is no term left on attribute $i$).

```
Function getAUtilityOrder(X̂) :
    R_û ← ∅;
    next ← N;
    For k ∈ N do
        h_k ← 1;
    end For
    While (next ≠ ∅) do
        Select randomly i ∈ next;
        R_û ← (R_û, u_{i,h_i});
        If (h_i ≠ n_i) then
            h_i ← h_i + 1;
        else
            next ← next \ {i};
        end If
    done
    return R_û;
End
```

Algorithm 1: Algorithm to generate a valid rank of all utility values.

Once $R_{\widehat{u}}$ is generated, the next step consists in associating ordered values between 0 and 1 for each utility value. As 0 and 1 are specific values, we calculate the maximum number of each according to the ranked utility values, based on those rules: an attribute $k$ with few number of distinct values ($n_k \leq 5$) has at the most one 0 and one 1; an attribute $k$ with more values ($5 \leq n_k \leq 10$) has at the most two 0 and two 1, etc. After that, with $n_R$ denoting the number of values in $R_{\widehat{u}}$ that are not equal to 0 or 1, we divide the segment $[0, 1]$ into $n_R$ segments of the same length, and a random value in this segment is attributed to each utility value. The function is **getUtilityValues**($\mathbf{R}_{\widehat{\mathbf{u}}}$) and its outcome is $\{u_{k,i} \ : \ k \in N \ , \ i \in \{1, \ldots, n_k\}\}$. It is noted $\widehat{u}$.

### 4.2. Fixed-point algorithm to obtain compatible weights and utilities

Once we have an initial value of $\widehat{u}$, we can launch the fixed-point algorithm that has to alternatively find Möbius coefficients that are relevant according to the previous set of utilities, and to find utilities that are relevant according to the previous set of Möbius coefficients. A relevant set of Möbius coefficients or utilities can produce a result that does not validate all the constraints. The objective of the algorithm is to return the solution that maximizes the number of learning examples that are satisfied. Our objective is to find a couple of Möbius coefficients and utilities that fulfill all the constraints.

#### *4.2.1. Find Möbius coefficients given the utility functions*

We assume here that $\widehat{u}$ is known and we wish to find $\widehat{m}$ that best fits the preferential information. As one may not satisfy the learning examples, we

introduce a binary slack variable for each learning example in the set $\{0,1\}$.

Constraint (12) (with $(x, V_x) \in \mathcal{P}$) is relaxed in the following way

$$\sum_{k\in N} m_k\, u_k(x_k) + \sum_{\{j,k\}\subseteq N} m_{j,k}\, u_j(x_j) \wedge u_k(x_k) - E_{x,V_x} \leq V_x \tag{17}$$

$$\sum_{k\in N} m_k\, u_k(x_k) + \sum_{\{j,k\}\subseteq N} m_{j,k}\, u_j(x_j) \wedge u_k(x_k) + E_{x,V_x} \geq V_x \tag{18}$$

where variable $E_{x,V_x} \in \{0,1\}$ is attached to the learning example $(x, V_x)$. The original constraint is satisfied if $E_{x,V_x} = 0$. Constraint (13) (with $(x, y) \in \mathcal{P}$) is relaxed in the following way

$$\sum_{k\in N} m_k \Big(u_k(x_k) - u_k(y_k)\Big) + \sum_{\{j,k\}\subseteq N} m_{j,k} \Big((u_j(x_j) \wedge u_k(x_k) - u_j(y_j) \wedge u_k(y_k)\Big) - E_{x,y} \leq 0 \tag{19}$$

where variable $E_{x,y} \in \{0,1\}$ is attached to the learning example $(x, y)$. The original constraint is satisfied if $E_{x,y} = 0$.

The value of $\widehat{m}$ is solution to the following problem, in which one minimizes the number of learning examples that are not satisfied:

$$\begin{array}{ll} \text{Minimize} & \displaystyle\sum_{(x,V_x)\in\mathcal{P}} E_{x,V_x} + \sum_{(x,y)\in\mathcal{P}} E_{x,y} \\ \text{under} & \left| \begin{array}{l} \widehat{m} \in \widehat{\mathcal{M}} \text{ satisfying (7) and (8)} \\ \forall (x,V_x) \in \mathcal{P}\ E_{x,V_x} \in \{0,1\} \\ \forall (x,V_x) \in \mathcal{P}\ \ \text{(17) and (18) hold} \\ \forall (x,y) \in \mathcal{P}\ E_{x,y} \in \{0,1\} \\ \forall (x,y) \in \mathcal{P}\ \ \text{(19) holds} \end{array} \right. \end{array}$$

As $\widehat{u}$ is fixed, the previous problem is linear in $\widehat{m}$. We denote by

$$compM(\widehat{u})$$

the pair $(\widehat{m}, f)$ where $\widehat{m}$ is the solution the previous linear program and $f$ is the corresponding value of the functional.

#### *4.2.2. Find utility functions given the Möbius coefficients*

We want here to find $\widehat{u}$ using $\widehat{m}$ obtained previously.

Constraints (12) and (13) use the Choquet integral and contain thus a minimum function between pairs of utilities. In order to linearize this term we introduce for each $x_j \in \widehat{X}_j$, $x_k \in \widehat{X}_k$ a new variable $u_{j,k}(x_j, x_k)$. In order to have $u_{j,k}(x_j, x_k)$ equal to $u_j(x_j) \wedge u_k(x_k)$, we introduce the following constraints:

$$u_{j,k}(x_j, x_k) - u_j(x_j) \leq 0 \tag{20}$$

$$u_{j,k}(x_j, x_k) - u_k(x_k) \leq 0 \tag{21}$$

$$u_{j,k}(x_j, x_k) - u_j(x_j) + \epsilon_{j,k}(x_j, x_k) \geq 0 \tag{22}$$

$$u_{j,k}(x_j, x_k) - u_k(x_k) - \epsilon_{j,k}(x_j, x_k) \geq -1 \tag{23}$$

where variable $\epsilon_{j,k}(x_j, x_k) \in \{0,1\}$. Relations (20) and (21) imply that $u_{j,k}(x_j, x_k) \leq \min(u_j(x_j), u_k(x_k))$. Relations (22) and (23) imply that either $u_{j,k}(x_j, x_k) \geq u_j(x_j)$ (when $\epsilon_{j,k}(x_j, x_k) = 0$) or $u_{j,k}(x_j, x_k) \geq u_k(x_k)$ (when $\epsilon_{j,k}(x_j, x_k) = 1$). Hence $\epsilon_{j,k}(x_j, x_k)$ is equal to 1 when $u_{j,k}(x_j, x_k) = u_k(x_k)$ and equal to 0 when $u_{j,k}(x_j, x_k) = u_j(x_j)$.

As in Section 4.2.1, we uses a binary slack variable for each learning example in the set $\{0,1\}$. Constraint (12) (with $(x, V_x) \in \mathcal{P}$) is relaxed in the following way

$$\sum_{k\in N} m_k\, u_k(x_k) + \sum_{\{j,k\}\subseteq N} m_{j,k}\, u_{j,k}(x_j, x_k) - E_{x,V_x} \leq V_x \tag{24}$$

$$\sum_{k\in N} m_k\, u_k(x_k) + \sum_{\{j,k\}\subseteq N} m_{j,k}\, u_{j,k}(x_j, x_k) + E_{x,V_x} \geq V_x \tag{25}$$

where variable $E_{x,V_x} \in \{0,1\}$ is attached to the learning example $(x, V_x)$. The original constraint is satisfied if $E_{x,V_x} = 0$. Constraint (13) (with $(x, y) \in \mathcal{P}$) is relaxed in the following way

$$\sum_{k\in N} m_k \Big(u_k(x_k) - u_k(y_k)\Big) + \sum_{\{j,k\}\subseteq N} m_{j,k} \Big((u_{j,k}(x_j, x_k) - u_{j,k}(y_j, y_k)\Big) - E_{x,y} \leq 0 \tag{26}$$

where variable $E_{x,y} \in \{0,1\}$ is attached to the learning example $(x, y)$. The original constraint is satisfied if $E_{x,y} = 0$.

The value of $\widehat{u}$ is solution to the following problem, in which one minimizes the number of learning examples that are not satisfied:

$$\begin{array}{ll} \text{Minimize} & \displaystyle\sum_{(x,V_x)\in\mathcal{P}} E_{x,V_x} + \sum_{(x,y)\in\mathcal{P}} E_{x,y} \\ \text{under} & \left| \begin{array}{l} \widehat{u} \in \widehat{\mathcal{U}} \text{ satisfying (1)} \\ \forall (x,V_x) \in \mathcal{P}\ E_{x,V_x} \in \{0,1\} \\ \forall (x,V_x) \in \mathcal{P}\ \ \text{(24) and (25) hold} \\ \forall (x,y) \in \mathcal{P}\ E_{x,y} \in \{0,1\} \\ \forall (x,y) \in \mathcal{P}\ \ \text{(26) holds} \\ \forall (x) \in \widehat{X}\ \epsilon_{j,k}(x_j, x_k) \in \{0,1\} \\ \forall (x) \in \widehat{X}\ \ \text{(20), (21), (22) and (23) hold} \end{array} \right. \end{array}$$

As $\widehat{m}$ is fixed, the previous problem is linear in $\widehat{u}$. We denote by

$$compU(\widehat{m})$$

the pair $(\widehat{u}, f)$ where $\widehat{u}$ is the solution the previous linear program and $f$ is the corresponding value of the functional.

### 4.3. General algorithm

We have introduced all necessary ingredients to show the general fixed-point algorithm. Algorithm

2 is the preference learning algorithm related to Choquet integral.

In this algorithm, integer $j$ is the global number of times a fixed-point loop is used. Integer $i$ is the number of iteration in the fixed-point sub-algorithm. Within each iteration of the fixed-point algorithm, functions $compU(\widehat{m}^{j,i})$ and $compM(\widehat{u}^{j,i})$ are launched. They return in particular the value of the functional $f$. It is equal to the number of non validated constraints. When $f = 0$, all learning constraints can be fulfilled and thus the algorithm can stop. In the opposite case, the algorithm continues. The algorithm is implemented in Java, and the problems $compU(\widehat{m}^{j,i})$ and $compM(\widehat{u}^{j,i})$ are based on the linear program solver LP solve [18].

```
Function PLAlgorithmWithChoquet() :
    While (j < maxIter) do
        R^j_u ← getAUtilityOrder(X̂);
        i = 1;
        û^{j,i} ← getUtilityValues(R^j_u);
        stop = False;
        While (¬stop) do
            (m̂^{j,i}, f) ← compM(û^{j,i});
            If (f = 0) then
                stop ← true;
            end If
            i ← i + 1;
            (û^{j,i}, f) ← compU(m̂^{j,i−1});
            If (f = 0 OR û^{j,i} = û^{j,i−1}) then
                stop ← true;
            end If
        done
        j ← j + 1;
    done
End
```

Algorithm 2: Algorithm to learn preference with Choquet Integral.

Let us end this section by a remark on convergence of the fixed-point. At each iteration, the aim is to minimize the number of training data that is not satisfied. Suppose that Suppose that, at a given iteration of the general algorithm (see Section 4.3), we have just run the optimization problem of Section 4.2.1 and the outcome is $compM(\widehat{u}^{k-1}) = (\widehat{m}^k, f^k)$, where $\widehat{m}^k$ are the Möbius coefficients, $f^k$ is the minimal number of learning examples that cannot be satisfied and $\widehat{u}^{k-1}$ is the value of the utilities found at the previous iteration. At the next iteration, the solution of the optimization problem of Section 4.2.2 is $compU(\widehat{m}^k) = (\widehat{u}^{k+1}, f^{k+1})$. By construction, $\widehat{u}^k$ satisfies the optimization problem of Section 4.2.2 with value $f^k$ for the functional. Hence one shall have $f^{k+1} \leq f^k$. And so on. This proves that the sequence of the values of the functional is non-negative and decreasing, and is thus converging. This does not necessarily mean that it converges to zero.

## 5. Illustration with an example

### 5.1. Description of the example

Let us describe a standard example used to motivate the Choquet integral with respect to capacities or bi-capacities [19]. The director of a university decides on students who are applying for graduate studies in management where some prerequisites from school are required. Students are indeed evaluated according to mathematics (M), statistics (S) and language skills (L). All the marks with respect to the scores are given on the same scale from 0 to 20. These three criteria serve as a basis for a pre-selection of the candidates. Let us consider the following four students.

| | $M$ | $S$ | $L$ |
|---|---|---|---|
| student $A$ | 14 | 17 | 6 |
| student $B$ | 14 | 15 | 8 |
| student $C$ | 9 | 17 | 6 |
| student $D$ | 9 | 15 | 8 |

The applicants have generally speaking a strong scientific background so that mathematics and statistics have a large importance to the director. However, he does not wish to favour too much students that have a scientific profile with some flaws in languages. Besides, mathematics and statistics are in some sense *redundant*, since, usually, students good at mathematics are also good at statistics. As a consequence, for students good in mathematics, the director prefers a student good at languages to one good at statistics. Student $A$ is penalized by his performance in languages. Hence, the director would prefer a student (with the same mark in mathematics) that is a little bit better in languages even if the student would be a little bit worse in statistics. This means that the director prefers $B$ to $A$:

$$A \prec B \tag{27}$$

Consider now a student that has a weakness in mathematics. In this case, since the applicants are supposed to have strong scientific skills, a student good in statistics is now preferred to one good in languages. Following above arguments, $C$ is preferred to $D$ even though $C$ has poor language skills:

$$C \succ D \tag{28}$$

As all attributes correspond to the same scale $[0, 20]$, one might think of applying the same utility function on them. We just need to divide the marks by 20 to obtain a satisfaction degree in $[0, 1]$:

$$\forall \in \{1, 2, 3\} \quad u_i(x_i) = \frac{x_i}{20}. \tag{29}$$

Let us indeed try to model (27) and (28) with the help of the Choquet integral when normalization (29) is applied. We have $H(A) = \frac{1}{20}\big(7 + 7\mu(\{M,S\}) + 2\mu(\{S\})\big)$ and $H(B) = \frac{1}{20}\big(8 + 6\mu(\{M,S\}) + \mu(\{S\})\big)$. This shows that (27) is equivalent to

$$\mu(\{M,S\}) + \mu(\{S\}) < 1.$$

Similarly, relation (28) is equivalent to $\mu(\{M,S\}) + \mu(\{S\}) > 1$, which contradicts previous inequality. Hence, the Choquet integral cannot model (27) and (28) with the simple normalization condition (29).

It is no surprise that the Choquet integral cannot model both (27) and (28) when (29) is imposed. This is due to the fact that the Choquet integral satisfies comonotonic additivity [20]. In our example, the marks of the four students $A$, $B$, $C$ and $D$ are ranked in the same way (under the simple normalization condition (29)): languages is the worst score, mathematics is the second best score, and statistics is the best score. Those four students are comonotonic.

In order to represent (27) and (28), we need to allow using different utility functions on each attribute. This makes sense in our example since each professor may apply different notation techniques so that 12 in mathematics may not have the same meaning as 12 in languages.

### 5.2. Result with our preference learning algorithm

We apply our algorithm to this example. After 4 iterations of the main loop, we obtain the following utility functions:

$$\begin{array}{ll} u_M(14) = 0.995 & u_M(9) = 0.137 \\ u_S(17) = 0.294 & u_S(15) = 0.137 \\ u_L(8) = 0.137 & u_L(6) = 0 \end{array}$$

Here are the evaluations on the three subjects and the global evaluation for each student: $(\{u_M, u_S, u_L\} \Rightarrow H(X))$ :

- Student A $= \{0.995; 0.294; 0\} \Rightarrow 0.632$
- Student B $= \{0.995; 0.137; 0.137\} \Rightarrow 0.682$
- Student C $= \{0.137; 0.294; 0\} \Rightarrow 0.187$
- Student D $= \{0.137; 0.137; 0.137\} \Rightarrow 0.137$

We also obtain the following Möbius coefficients:

| $m_M$ | $m_S$ | $m_L$ | $m_{M,S}$ | $m_{M,L}$ | $m_{S,L}$ |
|---|---|---|---|---|---|
| 0.635 | 0.635 | 0.001 | -0.635 | 0 | 0.364 |

The aim of the algorithm is not to find the most appropriate values of utility functions. It only tries to check whether there exist values of the parameters which fulfil the learning examples.

During the 4 iterations, the generated orders among the utilities were:

- Iteration 1: $u_L(6) < u_M(9) < u_L(8) < u_M(14) < u_S(15) < u_S(17)$
- Iteration 2: $u_M(9) < u_S(15) < u_L(6) < u_L(8) < u_M(14) < u_S(17)$
- Iteration 3: $u_L(6) < u_L(8) < u_M(9) < u_S(15) < u_M(14) < u_S(17)$
- Iteration 4: $u_L(6) < u_L(8) < u_M(9) < u_M(14) < u_S(15) < u_S(17)$

The order of the results is the following:

- Results: $u_L(6) < u_L(8) = u_M(9) = u_S(15) < u_S(17) < u_M(14)$

We can observe that this order is different from the last initialization order, at the iteration 4. It shows that the fixed-point algorithm tests various non-strict orders from each initial order. In this example, the number of iterations for each fixed-point loop was between 2 and 6.

## 6. Limits and improvements

As presented in 2.3, the order of the utility values related to criteria with interaction is significant. Therefore, the learning of the Möbius coefficients is sensitive to the values of the utility functions and in particular the order $\trianglerighteq$ among the values of the attributes. It is then important to launch the fixed-point algorithm from different orders $\trianglerighteq$. Yet the number of possible orders $\trianglerighteq$ is very large. For example, a simple case with three attributes and 2 distinct values on each produces 90 possible orders. In some real problems, the number of possible orders is too large to be tested in a reasonable maximal number of iterations with our algorithm. A suggestion to improve this is to use evolutionary algorithm to explore the set of possible orders $\trianglerighteq$ and converge more easily to potential relevant orders. The evaluation of each order $R_{\widehat{u}}$ can be provided by the result of $compM(\widehat{u})$, where 0 is the value of the best utility order, -1 is the value of the worst utility order, and for positive values the satisfaction decreases. From our experiment, we have noted that the final order after the execution of the fixed-point algorithm is different from the initial one. This suggests that it is not necessary to launch the fixed-point algorithm with all possible orders $\trianglerighteq$. Our future research will investigate this point more deeply.

We will also address the complexity issue of the learning approach, and in particular the impact of the number of learning examples, the number of criteria, the kind of interactions among criteria and the number of elements on the attributes.

The aim of this work is only to find values of the utilities and of the capacity that fulfils the holistic learning examples. However, these values may not be suitable to generate recommendations for new alternatives. For future research, we will also generate values of the parameters (starting from the same order $\trianglerighteq$) that can be used to generate recommendations for the user.

## 7. Conclusion

The algorithm presented in this paper aims at simplifying the task of building a multi-criteria decision model by using several preferences or evaluations, instead of describing the utility on each attribute and the importance of each attribute through the Möbius coefficients. The objective is to find a couple of utilities and Möbius coefficients $(\widehat{u}, \widehat{m})$ that fulfills all the constraints of monotonicity, normalization and the constraints related to the learning examples. As described, the problem is bilinear. To try to find a solution, we propose to use a fixed-point algorithm to alternatively find a $\widehat{m}$ according to a $\widehat{u}$ (denoted $compU(\widehat{m})$), and find a $\widehat{u}$ according to a $\widehat{m}$ (denoted $compM(\widehat{u})$). Each problem can provide a partial solution that does not validated all the constraints. This partial solution is then used in the next problem to see if the change of variables can provide a better solution. The solution is obtained when all the constraints are validated with a couple $(\widehat{u}, \widehat{m})$.

## References


[1] J. Figueira, S. Greco, and M. Ehrgott, editors. *Multiple Criteria Decision Analysis: State of the Art Surveys*. Kluwer Acad. Publ., 2005.

[2] D.H. Krantz, R.D. Luce, P. Suppes, and A. Tversky. *Foundations of measurement*, volume 1: Additive and Polynomial Representations. Academic Press, 1971.

[3] M. Grabisch. The application of fuzzy integrals in multicriteria decision making. *European J. of Operational Research*, 89:445–456, 1996.

[4] M. Grabisch and Ch. Labreuche. A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid. *Annals of Operation Research*, 175:247–286, 2010.

[5] Ch. Labreuche and M. Grabisch. The Choquet integral for the aggregation of interval scales in multicriteria decision making. *Fuzzy Sets & Systems*, 137:11–26, 2003.

[6] M. Grabisch, I. Kojadinovic, and P. Meyer. A review of capacity identification methods for Choquet integral based multi-attribute utility theory — applications of the Kappalab R package. *Eur. J. of Operational Research*, 186:766–785, 2008.

[7] C. A. Bana e Costa and J.-C. Vansnick. A theoretical framework for Measuring Attractiveness by a Categorical Based Evaluation TecHnique (MACBETH). In *Proc. XIth Int. Conf. on MultiCriteria Decision Making*, pages 15–24, Coimbra, Portugal, August 1994.

[8] D. Bouyssou, M. Couceiro, C. Labreuche, J.-L. Marichal, and B. Mayag. Using choquet integral in machine learning: what can MCDA bring? In *Workshop from Multiple Criteria Decision Aid to Preference Learning*, Mons, Belgium, November 15-16 2012.

[9] Ch. Labreuche. Construction of a Choquet integral and the value functions without any commensurateness assumption in multi-criteria decision making. In *Int. Conf. Of the Euro Society for Fuzzy Logic and Technology (EUSFLAT)*, Aix Les Bains, France, July 18-22 2011.

[10] Ch. Labreuche. An axiomatization of the Choquet integral and its utility functions without any commensurateness assumption. In *Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, Catania, Italy, July 9-13 2012.

[11] S. Angilella, S. Greco, F. Lamantia, and B. Matarazzo. Assessing non-additive utility for multicriteria decision aid. *European Journal of Operational Research*, 158:734–744, 2004.

[12] J. Fürnkranz and E. Hüllermeier, editors. *Preference Learning*. Springer-Verlag, 2010.

[13] R. L. Keeney and H. Raiffa. *Decision with Multiple Objectives*. Wiley, New York, 1976.

[14] G. Choquet. Theory of capacities. *Annales de l'Institut Fourier*, 5:131–295, 1953.

[15] M. Sugeno. *Theory of fuzzy integrals and its applications*. PhD thesis, Tokyo Institute of Technology, 1974.

[16] G. C. Rota. On the foundations of combinatorial theory I. Theory of Möbius functions. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 2:340–368, 1964.

[17] M. Grabisch. $k$-order additive discrete fuzzy measures and their representation. *Fuzzy Sets and Systems*, 92:167–189, 1997.

[18] M. Berkelaar, K. Eikland, and P. Notebaert. LP solve: Open source (mixed-integer) linear programming system. Technical report, Version 5.5 dated May 16, Rotterdam, The Netherlands, 2005.

[19] M. Grabisch and Ch. Labreuche. Fuzzy measures and integrals in MCDA. In M. Ehrgott J. Figueira, S. Greco, editor, *Multiple Criteria Decision Analysis - State of the Art Surveys*, pages 563–608. Springer's International Series, 2005.

[20] D. Schmeidler. Integral representation without additivity. *Proc. of the Amer. Math. Soc.*, 97(2):255–261, 1986.