Extracting Decision Rules from Qualitative Data via Sugeno Utility Functionals

Quentin Brabant, Miguel Couceiro, Didier Dubois, Henri Prade, Agnès Rico

Extracting Decision Rules from Qualitative Data via Sugeno Utility Functionals

Quentin Brabant*[1], Miguel Couceiro[1],
Didier Dubois[2], Henri Prade[2], Agnès Rico[3]

(1) LORIA, (Inria Nancy Grand Est & Université de Lorraine),
Campus scientifique, BP 239, 54506 Vandoeuvre-ls-Nancy

(2) IRIT, CNRS & Université Paul Sabatier,
118 route de Narbonne 31062 Toulouse

(3) ERIC & Université Claude Bernard Lyon 1,
43 bld du 11-11, 69100 Villeurbanne

e-mails: quentin.brabant@loria.fr, miguel.couceiro@loria.fr,
dubois@irit.fr, prade@irit.fr, agnes.rico@univ-lyon1.fr

Abstract

Sugeno integrals are qualitative aggregation functions. They are used in multiple criteria decision making and decision under uncertainty, for computing global evaluations of items, based on local evaluations. The combination of a Sugeno integral with unary order preserving functions on each criterion is called a Sugeno utility functionals (SUF). A noteworthy property of SUF is that they represent multi-threshold decision rules, while Sugeno integrals represent single-threshold ones. However, not all sets of multi-threshold rules can be represented by a single SUF. In this paper, we consider functions defined as the minimum or the maximum of several SUF. These max-SUF and min-SUF can represent all functions that can be described by a set of multi-threshold rules, i.e., all order-preserving functions on finite scales. We study their potential advantages as a compact representation of a big set of rules, as well as an intermediary step for extracting rules from empirical datasets.

1 Introduction

Sugeno integrals [12] are aggregation functions that are used in multiple criteria decision making and in decision under uncertainty [7, 9]. They are qualitative aggregation functions because they can be defined on non-numerical scales (more precisely on distributive lattices [5]). In this paper we only consider Sugeno integrals defined on completely ordered scales, and denote such a scale by L. A noteworthy property of Sugeno integrals is that their output is always comprised between the minimum and the maximum of their parameters. Moreover, Sugeno integrals on L are a subclass of lattice polynomials on L. More precisely, they correspond to all idempotent functions from

L^n to L that can be formulated using min and max operations, variables and constants. From a decision making point of view, they also can be regarded as functions whose result depends on an importance value assigned to each subset of criteria. Sugeno integrals are known to represent any set of single-threshold if-then rules [8, 10] of the form

$$x_1 \geq \alpha \text{ and } x_2 \geq \alpha \ldots \text{ and } x_n \geq \alpha \Rightarrow y \geq \alpha \text{ (selection rules), or}$$
$$x_1 \leq \alpha \text{ and } x_2 \leq \alpha \ldots \text{ and } x_n \leq \alpha \Rightarrow y \leq \alpha \text{ (deletion rules).}$$

Sugeno utility functionals (SUF) are a generalization of Sugeno integrals [6] where each criterion value is mapped to an element of L by an order preserving function. They allow to represent multi-threshold rules of the form:

$$x_1 \geq \alpha_1 \text{ and } x_2 \geq \alpha_2 \ldots \text{ and } x_n \geq \alpha_n \Rightarrow y \geq \delta \text{ (selection rules), or}$$
$$x_1 \leq \alpha_1 \text{ and } x_2 \leq \alpha_2 \ldots \text{ and } x_n \leq \alpha_1 \Rightarrow y \leq \delta \text{ (deletion rules).}$$

However, although any *single* multi-threshold rule can be represented by a SUF, not all *sets* of multi-threshold rules can be represented by a single SUF [3].

In this paper, we consider functions defined as disjunctions or conjunctions of SUF, recently introduced in [3]. They capture all order-preserving piecewise unary functions on finite scales, this is to say, all functions that can be represented by means of a set of multi-threshold rules. We study the potential advantages of this representation based on combinations of SUF. In particular, we investigate whether it is possible to represent a large set of rules by means of only a few SUF, and whether combinations of SUF can help learning models which offer a good trade-off between simplicity and predictive accuracy.

In the next section we present combinations of SUF as a framework with equivalent expressivity to that of decision rules. In Section 3, we deal with the problem of finding a minimal combination of SUF that represents a set of rules. In Section 4, we propose a method for approximately representing real datasets by means of a disjunction of SUF. We show that it achieves predictive accuracy scores similar to the rule sets learned by the rough set-based method VC-DomLEM [1]. We also look at the compactness of the obtained model, and at the relation between the compactness of the model and its predictive accuracy. We omit the proof of most results because of space limitation.

2 Preliminaries

We use the terminology of multiple criteria decision-making where some objects are evaluated according to several criteria. We denote by $C = \{1, \ldots, n\}$ a set of criteria, by 2^C its power set, and by $X_1, \ldots, X_n$ and L totally ordered scales with top $1_{X_1}, \ldots, 1_{X_n}$ and 1_L and bottom $0_{X_1}, \ldots, 0_{X_n}$ and 0_L, respectively. We denote by ν the order reversing operation on L (ν is involutive and such that $\nu(0) = 1$ and $\nu(1) = 0$. We denote the Cartesian product $X_1 \times \cdots \times X_n$ by $\mathbf{X}$. An object is represented by a vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbf{X}$ where x_i is the evaluation of x w.r.t. criterion i. Let $f : \mathbf{X} \to L$ be an evaluation function.

2.1 Decision Rules

Order-preserving functions from $\mathbf{X}$ to L can always be described by a set of selection rules or of deletion rules [3]. For the sake of simplicity, in this paper we mainly focus on selection rules. When

there is no risk of ambiguity, we simply refer to selection rules as rules. Since any selection rule $r \in R$ has the form

$$x_1 \geq \alpha_1^r, \ldots, x_n \geq \alpha_n^r \Rightarrow f(\mathbf{x}) \geq \delta^r,$$

we will use the following abbreviation for defining a rule

$$r : \alpha_1^r, \ldots, \alpha_n^r \Rightarrow \delta^r.$$

Moreover, the left hand side of a rule r will be denoted by $\boldsymbol{\alpha}^r = (\alpha_1^r, \ldots, \alpha_n^r)$. We say that a function $f : \mathbf{X} \to L$ is *compatible* with a selection rule r if $f(\mathbf{x}) \geq \delta^r$ for all $\mathbf{x}$ such that $x_i \geq \alpha_i^r$ for each $i \in C$. For any selection rule r, we will denote by f_r the least function from $\mathbf{X}$ to L that is compatible with r, i.e., the function defined by

$$f_r(\mathbf{x}) = \delta^r \text{ if } [x_i \geq \alpha_i^r \ \forall i \in C], \ 0 \text{ otherwise,}$$

for all $x \in \mathbf{X}$. We say that a criterion $i \in C$ is *active* in r if $\alpha_i^r > 0_{X_i}$. Moreover, we denote by A^r the set of criteria active in a rule r. For any set of rules R, we denote by f_R the least function compatible with all rules in R, defined by

$$f_R = \max_{r \in R} f_r(\mathbf{x})$$

for all $\mathbf{x} \in \mathbf{X}$. We say that a function $f : \mathbf{X} \to L$ *represents* a set of selection rules R (or equivalently, that R represents f) if $f = f_R$. We say that a set of selection rules is *redundant* if there exists $r, s \in R$ such that $r \neq s$,

$$\alpha_i^r \geq \alpha_i^s \text{ for all } i \in C \quad \text{and} \quad \delta^s \geq \delta^r, \tag{1}$$

or, equivalently, if there exists $r \in R$ such that $f_R = f_{R \setminus \{r\}}$. A set of rules that is not redundant is said to be *irredundant*. We define the *equivalence class* of a set of rules R as $[R] = \{R' \mid f_R = f_{R'}\}$.

Lemma 1 *The equivalence class of R has one minimal element, which is*

$$R^{\min} = \{r \in R \mid \nexists s \in R : f_r \leq f_s\} = \bigcap_{R' \in [R]} R', \tag{2}$$

and is the only irredundant element of $[R]$.

2.2 Sugeno Integrals and Their Generalizations

A Sugeno integral is defined with respect to a capacity which is a set function $\mu : 2^C \to L$ that satisfies $\mu(\emptyset) = 0$, $\mu(C) = 1$ and $\mu(I) \leq \mu(J)$ for all $I \subseteq J \subseteq C$. The conjugate capacity of μ is defined by $\mu^c(I) = \nu(\mu(I^c))$, where I^c is the complement of I. The capacity can be seen as a function assigning an importance level to each subset of criteria. The Sugeno integral S_μ associated to μ can be defined in two ways:

$$S_\mu(\mathbf{x}) = \max_{I \subseteq C} \min(\mu(I), \min_{i \in I}(x_i)) = \min_{I \subseteq C} \max(\mu(I^c), \max_{i \in I}(x_i)),$$

for all $\mathbf{x} \in L^n$ [11]. The inner qualitative Möbius transform of a capacity μ is a mapping $\mu_\# : 2^C \to L$ defined by

$$\mu_\#(I) = \mu(I) \text{ if } \mu(I) > \max_{J \subset I} \mu(J), \text{ and 0 otherwise.}$$

A set $I \subseteq C$ such that $\mu(I) > 0$ is called a focal set. The set of focal sets of μ is denoted by $\mathcal{F}(\mu)$. Since $\mu(A) = \max_{I \subseteq A} \mu_{\#}(I)$ for all $A \subseteq C$, the set function $\mu_{\#}$ contains the minimal amount of information needed to reconstruct μ. The qualitative Möbius transform provides a concise representation of the Sugeno integral as:

$$S_\mu(\mathbf{x}) = \max_{I \in \mathcal{F}(\mu)} \min(\mu_{\#}(I), \min_{i \in I}(x_i)), = \min_{I \in \mathcal{F}(\mu^c)} \max(\nu(\mu^c_{\#}(I)), \max_{i \in I}(x_i)).$$

A Sugeno utility functional (SUF) is a combination of a Sugeno integral and unary order preserving maps on each criterion [3, 10]. Formally, a SUF is a function $S_{\mu,\varphi}$ defined by

$$S_{\mu,\varphi}(\mathbf{x}) = \max_{I \in \mathcal{F}(\mu)} \min(\mu(I), \min_{i \in I} \varphi_i(x_i)), \quad \text{for all } \mathbf{x} \in \mathbf{X},$$

where μ is a capacity, $\varphi = (\varphi_1, \ldots, \varphi_n)$ and, for all $i \in C$, $\varphi_i : X_i \to L$ is order preserving, with $\varphi(0_{X_i}) = 0_L$ and $\varphi(1_{X_i}) = 1_L$.

It is shown in [10] that any SUF can be represented in terms of single-thresholded rules of the form "$\varphi_1(x_1) \geq \alpha, \ldots, \varphi_n(x_n) \geq \alpha \Rightarrow f(\mathbf{x}) \geq \alpha$" (in other words, for all SUF $S_{\mu,\varphi}$ there is a set of rules R such that $f_R = S_{\mu,\varphi}$). On the contrary, some sets of rules cannot be represented by a SUF. This justifies the use of combinations of SUF. A max-SUF (resp. min-SUF) is defined by

$$f(\mathbf{x}) = \max_{i \in \{1,\ldots,k\}} \underline{S}^i(\mathbf{x}) \quad \left(\text{resp. } f(\mathbf{x}) = \min_{j \in \{1,\ldots,\ell\}} \overline{S}^j(\mathbf{x}) \right),$$

for all $\mathbf{x} \in \mathbf{X}$, where $\underline{S}^1, \ldots, \underline{S}^k, \overline{S^1}, \ldots \overline{S}^\ell$ are SUF.

Remark 1 *It is shown in [3] that min-SUF and max-SUF can represent any set of deletion or selection rules, respectively. In other words, any order-preserving function from $\boldsymbol{X}$ to L can be expressed by a max-SUF and by a min-SUF. Also, since any SUF is such that $S_{\mu,\varphi}(0_{X_1}, \ldots, 0_{X_n}) = 0_L$ and $S_{\mu,\varphi}(1_{X_1}, \ldots, 1_{X_n}) = 1_L$, note that no combination of SUF from $\boldsymbol{X}$ to L can represent a set of rules R such that $f_R(0_{X_1}, \ldots, 0_{X_n}) > 0_L$ or $f_R(1_{X_1}, \ldots, 1_{X_n}) < 1_L$. However, one can take $L' = \{y \in L \mid \exists \boldsymbol{x} \in X : f_R(\boldsymbol{x}) = y\}$ and find a combination of SUF from $\boldsymbol{X}$ to L' that represents R.*

Based on these results, we can try to model any order-preserving function defined on finite scales by means of max-SUF or min-SUF. Moreover, datasets that can be conveniently approximated by means of rules could also be approximated by means of a max-SUF or min-SUF. In the next section, we focus on the exact representation of an order-preserving function.

3 Representing a Set of Rules by a (max-)SUF

In this section we will address the following problems. Given a set of rules R:

1. Determine whether R is *SUF-representable*, i.e., whether there is a SUF such that $f_R = S_{\mu,\varphi}$.
2. Find such a SUF if it exists.
3. In the case it does not exist, find a max-SUF that represents R while involving the least possible number of SUFs.

3.1 From a SUF to a Rule Set and Back

We already know that any SUF can be represented by a set of rules. Let us denote by $R_{\mu,\varphi}$ the set of rules defined from focal sets in $\mathcal{F}(\mu)$ as

$$\bigcup_{F\in\mathcal{F}(\mu)}\bigcup_{\delta^r\leq\mu(F)}\{r \mid A^r = F \text{ and } \forall i\in A^r : \alpha_i^r = \min\{x_i\in X_i \mid \varphi_i(x_i)\geq\delta^r\}\}.$$

As it will be shown in Lemma 2, this set of rules is equivalent to $S_{\mu,\varphi}$.

Although it is not always possible to find a SUF that represents a given set of rules, we can give a method for constructing such a SUF, when it exists. For any set of rules R, let $S_R^{\max}$ be the SUF defined by $S_R^{\max} = S_{\mu,\varphi}$, where $\mu(C) = 1_L$,

$$\forall I\subset C \text{ such that } I\neq\emptyset : \qquad \mu(I) = \max_{\substack{r\in R,\\ A^r\subseteq I}}\delta^r, \tag{3}$$

$$\forall i\in[n],\ \forall\mathbf{x}\in\mathbf{X} \text{ such that } x_i\neq 1_{X_i} : \qquad \varphi_i(x_i) = \max_{\substack{r\in R,\\ 0<\alpha_i^r\leq x_i}}\delta^r, \tag{4}$$

and $\varphi_i(1_{X_i}) = 1_L$ for all $i\in C$. From this definition, we easily see that we always have $S_R^{\max}\geq f_R$. However it can be the case that $S_R^{\max} > f_R$ as shown in the following example.

Example 1 *Consider a set of three criteria $C = \{1,2,3\}$, the scales $X_1 = X_2 = X_3 = L = \{0,a,b,1\}$, with $0<a<b<1$, and the rule set $R = \{r^1, r^2, r^3\}$, with*

$$r^1 : 0,b,1\Rightarrow 1, \quad r^2 : a,a,0\Rightarrow a, \quad r^3 : a,0,b\Rightarrow b.$$

Let $S_{\mu,\varphi} = S_R^{\max}$. The function μ is such that

$$\mu_\#(\{1,2\}) = a \quad \mu_\#(\{1,3\}) = b, \quad \mu_\#(\{2,3\}) = 1.$$

and, for all other $I\subset C$, $\mu_\#(I) = 0$. Moreover, we have

$$\varphi_1(a) = \varphi_1(b) = b, \quad \varphi_2(a) = a,\ \varphi_2(b) = 1, \quad \text{and} \quad \varphi_3(a) = 0,\ \varphi_3(b) = b.$$

One can check that $S_{\mu,\varphi}(0,b,b) = b$, while $f_R(0,b,b) = 0$.

We will show that, when $S_R^{\max} > f_R$, there exists no SUF that represents R (see Proposition 4).

Lemma 2 *For any SUF $S_{\mu,\varphi}$, we have $S_{\mu,\varphi} = f_{R_{\mu,\varphi}} = S_{R_{\mu,\varphi}}^{\max}$.*

Proof 1 *Let $S_{\mu,\varphi}$ be a SUF, and let μ^* and φ^* be such that $S_{R_{\mu,\varphi}}^{\max} = S_{\mu^*,\varphi^*}$. First, take any $\boldsymbol{x}\in\boldsymbol{X}$, $y\in L$ such that $S_{\mu,\varphi}(\boldsymbol{x})\geq y$. Necessarily, there is $F\in\mathcal{F}(\mu)$ such that*

$$\min(\mu(F), \min_{i\in F}\varphi_i(x_i))) \geq y.$$

Therefore $\mu(F)\geq y$ and $\forall i\in F : \varphi_i(x_i)\geq y$. From the definition of $R_{\mu,\varphi}$ it follows that there is $r\in R_{\mu,\varphi}$ such that $\delta^r = \mu(F)\geq y$, $A^r = F$ and $\forall i\in A^r : \alpha_i^r\leq x_i$. So we have $f_r(\boldsymbol{x})\geq y$ and thus $f_{R_{\mu,\varphi}}(\boldsymbol{x})\geq y$. Now, from the definition of $S_{R_{\mu,\varphi}}^{\max}$, it follows that $\mu^*(A^r)\geq y$ and $\forall i\in A^r : \varphi_i^*(x_i)\geq y$. Therefore $S_{R_{\mu,\varphi}}^{\max}\geq y$. Summing up, we have proven $S_{\mu,\varphi}\leq S_{R_{\mu,\varphi}}^{\max}$ and $S_{\mu,\varphi}\leq f_{R_{\mu,\varphi}}$.

Now we will prove $S_{\mu,\varphi} \geq f_{R_{\mu,\varphi}}$. *Let* $\boldsymbol{x} \in \boldsymbol{X}$, $y \in L$ *such that* $f_{R_{\mu,\varphi}}(\boldsymbol{x}) \geq y$. *Necessarily there is* $r \in R_{\mu,\varphi}$ *such that* $f_r(\boldsymbol{x}) \geq y$. *Therefore* $\delta^r \geq y$, *and from the definition of* $R_{\mu,\varphi}$ *it follows that* $\mu(A^r) \geq \delta^r$. *Moreover for all* $i \in A^r$ *we have*

$$x_i \geq \alpha_i^r = \min\{z_i \in X_i \mid \varphi_i(z_i) \geq \delta^r\},$$

and thus $\varphi(x_i) \geq \delta^r$. *Finally we get that for all* y *s.t.* $f_{R_{\mu,\varphi}}(\boldsymbol{x}) \geq y$:

$$S_{\mu,\varphi}(\boldsymbol{x}) \geq \min(\mu(A^r), \min_{i \in A^r} \varphi_i(x_i)) \geq \delta^r \geq y.$$

Therefore $S_{\mu,\varphi} \geq f_{R_{\mu,\varphi}}$.

We still have to prove $S^{\max}_{R_{\mu,\varphi}} \leq S_{\mu,\varphi}$. *Let* $\boldsymbol{x} \in \boldsymbol{X}$ *and* $y \in L$ *be such that* $S^{\max}_{R_{\mu,\varphi}}(\boldsymbol{x}) \geq y$. *Necessarily there is* $F \in \mathcal{F}(\mu^*)$ *such that*

$$\min(\mu^*(F), \min_{i \in F} \varphi_i^*(x_i)) \geq y \tag{5}$$

Thus $\mu^*(F) \geq y$ *and from the definition of* $S^{\max}_{R_{\mu,\varphi}}$ *it follows that there is* $r \in R$ *such that* $A^r \subseteq F$ *and* $\delta^r \geq y$. *From the definition of* $R_{\mu,\varphi}$ *we obtain that* $\mu(A^r) \geq y$. *From* (5) *we also get that* $\forall i \in F : \varphi_i^*(x_i) \geq y$. *Again, from the definition of* $S^{\max}_{R_{\mu,\varphi}}$, *it follows that, for each* $i \in F$, *there is* $r^i \in R_{\mu,\varphi}$ *such that* $\delta^{r^i} \geq y$ *and* $0 < \alpha_i^{r^i} \leq x_i$. *So, for each* $i \in F$:

$$x_i \geq \alpha_i^{r^i} = \min\{z_i \in X_i \mid \varphi_i(z_i) \geq \delta^{r^i}\},$$

and thus $\varphi_i(x_i) \geq \delta^{r^i}$. *Therefore we get*

$$\min(\mu(A^r), \min_{i \in A^r} \varphi_i(x_i)) \geq y.$$

We have shown that $S^{\max}_{R_{\mu,\varphi}} \leq S_{\mu,\varphi}$, *and the proof is complete.* □

Lemma 3 *Let* R *and* R' *be two sets of selection rules belonging to the same equivalence class. Necessarily* $S^{\max}_R = S^{\max}_{R'}$.

Lemma 4 *For any set of selection rules* R, *if there exists a SUF* S *such that* $S = f_R$, *then* $S = S^{\max}_R$.

Relying on Proposition 4, we are able to identify sets of rules that can be represented by a single SUF, and to define such a SUF if it exists. A simple procedure for doing so, starting from R, is to compute μ and φ such that $S^{\max}_R = S_{\mu,\varphi}$ and then to compute $R_{\mu,\varphi}$. If $R_{\mu,\varphi}$ and R are in the same equivalence class, then $S^{\max}_R$ is the SUF equivalent to R, otherwise there is no such SUF.

3.2 From a Rule Set to a max-SUF

Now consider the case where R cannot be represented by a SUF, but we want to find a max-SUF that involves the least possible number of SUF for representing R. In order to find such an "optimally parsimonious" max-SUF, one has to build the smallest partition (in terms of number of subsets) where each subset P is SUF-representable. All such partitions cannot be enumerated in reasonable time. Moreover note that, although in Example 1, r^1 and r^3 are responsible for the fact that $S^{\max}_R > f_R$, whether R is SUF-representable or not depends on larger combinations of rules. There are cases where each $A \subset R$ of size at most n is SUF-representable, while R is not. One example of such a case is the following.

Example 2 *Let $n = 4$. Consider $L = \{0, a, b, 1\}$ and the set of selection rules $R = \{r^1, \ldots, r^5\}$ with domain L^4 and codomain L, where*

$$r^1 : a, a, 0, 0 \Rightarrow b,$$
$$r^2 : a, 0, a, 0 \Rightarrow b,$$
$$r^3 : a, 0, 0, a \Rightarrow b,$$
$$r^4 : 0, a, a, a \Rightarrow a,$$
$$r^5 : 0, b, b, b \Rightarrow 1.$$

One can check that $S_R^{\max} > f_R$, while each subset of R of size 4 is SUF-representable. □

Algorithm 1 presents a greedy method for extracting the max-SUF representing a set of rules. Although the resulting max-SUF is not necessarily minimal, it constitutes a first approximation.

Algorithm 1: Builds a set of SUF $\boldsymbol{S}$ that represents a given set of rules R

```
P ← {}
for each r ∈ R do
    covered ← false
    for each P ∈ P do
        if f_{P∪{r}} = S^max_{P∪{r}} then
            add r to P
            covered ← true
            break foreach
    if covered = false then
        add {r} to P
S ← {S^max_P | P ∈ P}
```

Remark 2 *In Algorithm 1, one does not have to compute $S^{\max}_{P\cup\{r\}}$ from scratch each time a rule r is added to P; the function $S_P^{\max}$ can be updated iteratively.*

In order to get an idea of the number of SUFs required to represent a set of rules, we randomly generated sets of rules, and used Algorithm 1 to find their representations in terms of max-SUF.

Algorithm 2: Random generation of a set of rules R, for a given domain $\mathbf{X}$, codomain L, and real number $p << 1$

```
R ← {}
for i from 1 to |X| * p do
    pick a random α^r = (α^r_1, ..., α^r_n) ∈ X\D
    V = {δ^r ∈ L | such that (1) holds for all s ∈ R}
    if V ≠ ∅ then
        pick a random δ^r in V
        add the rule r : α^r_1, ..., α^r_n ⇒ δ^r to R
```

Our aim was to approximate the number of utility functionals necessary to represent a set of rules, depending on the size of the set. Algorithm 2 describes the random generation of a rule set. The number of rules depends on the number of criteria and on the size of L. For the sake of simplicity, for our test we set $\mathbf{X} = L^n$.

Table 1 displays the variation of the number of rules and Sugeno-utility functionals obtained, depending on n and the size of L. These results suggest that in general, the representation of a

Table 1: Number of rules/Sugeno-utility functionals

		Size of L						
		2	3	4	5	6	7	8
n	2	2/1	2/1	3/1	2/1	3/1	4/1	4/2
	3	2/1	3/2	4/1	5/3	7/4	9/5	13/7
	4	2/1	4/2	7/4	14/8	26/15	43/23	70/38
	5	2/1	5/3	20/11	50/28	112/60	248/122	445/209
	6	3/1	13/7	60/31	202/103	600/295	1397/642	3204/1384
	7	3/1	31/13	197/93	873/410	3125/1386	8942/3859	21762/8780

set of selection rules in terms of a max-SUF is not very compact, since many SUFs have to be involved. However, these results only concern randomly generated rule sets. In the next section, we use combination of SUFs for empirical study on real datasets.

4 Modeling Empirical Data

Formally, we represent a dataset by a multiset $\mathcal{D}$, whose elements belong to $\mathbf{X} \times L$. Elements of $\mathcal{D}$ are called *instances*, and values of L are referred to as *classes*. In this section we present a method for representing an empirical dataset by the mean of a max-SUF. This method is then applied on 12 datasets, the characteristics of which are given by Table 2. The process by which we extract a

Table 2: Description of the datasets. Further information can be found in [1].

Id	**Name**	**Instances**	**Criteria**	**Classes**	**Id**	**Name**	**Instances**	**Criteria**	**Classes**
1	breast-c	286	8	2	7	denbosch	119	8	2
2	breast-w	699	9	2	8	ERA	1000	4	9
3	car	1296	6	4	9	ESL	488	4	9
4	CPU	209	6	4	10	LEV	1000	4	5
5	bank-g	1411	16	2	11	SWD	1000	10	4
6	fame	1328	10	5	12	windsor	546	10	4

max-SUF from a dataset can be divided into four steps.

1. Select an order-preserving subset of data.
2. Associate a rule to each instance.
3. Group rules into a max-SUF.
4. Simplify the model by pruning some of the SUF that compose it.

Step 1. Selection of an order-preserving subset of data. Real life datasets often contain pairs of instances that are *order-reversing*, i.e, instances $(\mathbf{x}, y)$ and $(\mathbf{x'}, y')$ such that $\mathbf{x} \leq \mathbf{x'}$ and $y > y'$. In this step, we build a graph where the set of vertices is $\mathcal{D}$, and the set of edges contains each order-reversing pair of instances. Then, we iteratively remove from $\mathcal{D}$ the node with the largest number of neighbors, until no edge remains in the graph. In the next steps, we refer as $\mathcal{D}^-$ to the data remaining after Step 1. Table 3 presents the ratio of instances which are removed in each of the 12 datasets.

Table 3: Average ratio of data that is removed during Step 1, for each dataset.

1	2	3	4	5	6	7	8	9	10	11	12
.176	.008	.001	.0	.003	.025	.046	.657	.198	.333	.356	.237

Step 2. Associate a rule to each instance. A naive way to proceed would be to return the following set of rules: $R = \{r \mid \exists (\mathbf{x}, y) \in \mathcal{D}^- : [\delta^r = y \text{ and } \forall i \in C : \alpha_i^r = x_i]\}$. However, with most datasets, this would lead to all criteria being active in most rules. This is problematic because SUFs that will be learned by grouping such rules will only have focal sets of great size. Algorithm 3 provides an alternative solution, in which some criteria are set to 0 before rules are extracted. Its principle is that the i^{th} criterion value of an instance can be set to 0 if by doing so the data are still order-preserving. Since the order in which criteria are considered can influence the result of the algorithm, we define this order w.r.t. the discriminative power of each criterion in each instance. A rough estimation of this feature is given by the function $u : \bigcup_{i \in C} X_i \times L \to \mathbb{N}$ defined by

$$u(x_i, y) = \left|\{(\mathbf{x'}, y') \in \mathcal{D}^- \mid [y > y' \text{ and } x_i > x'_i] \text{ or } [y < y' \text{ and } x_i < x'_i]\}\right|.$$

Intuitively, for any $(\mathbf{x}, y) \in \mathcal{D}^-$, $u(x_i, y)$ represents the number of other instances $(\mathbf{x'}, y') \in \mathcal{D}^-$ such that the relation between y and y' could be explained (at least partially) by the i^{th} criterion.

Algorithm 3: Builds a set of rules R covering all instances of a given $\mathcal{D}^-$

```
1 R ← {}
2 for each (x, y) ∈ D⁻ do
3     for x_i ∈ x in ascending order of u(x_i, y) do
4         if D⁻ ∪ {((x_1, ..., x_{i-1}, 0_{X_i}, x_{i+1}, ..., x_n), y)} is order-preserving then
5             x_i ← 0_{X_i}
6     add x_1, ..., x_n ⇒ y to R
```

Step 3. Group rules into a max-SUF. In this step we simply exploit a variant of Algorithm 1, where the condition in line 6 is replaced by "$S^{\max}_{P \cup \{r\}}(\mathbf{x}) \leq y$ for all $(\mathbf{x}, y) \in \mathcal{D}^-$". This guarantees that the max-SUF obtained is greater than or equal to f_R and does not missclassify any instance of $\mathcal{D}^-$.

Step 4. Simplify the model. In this step, we remove some of the SUFs from the model (see Algorithm 4). The algorithm depends on a parameter $\rho \in [0, 1]$, usually set close to 1, which represents the ratio of accuracy that has to be preserved when removing a SUF from the model.

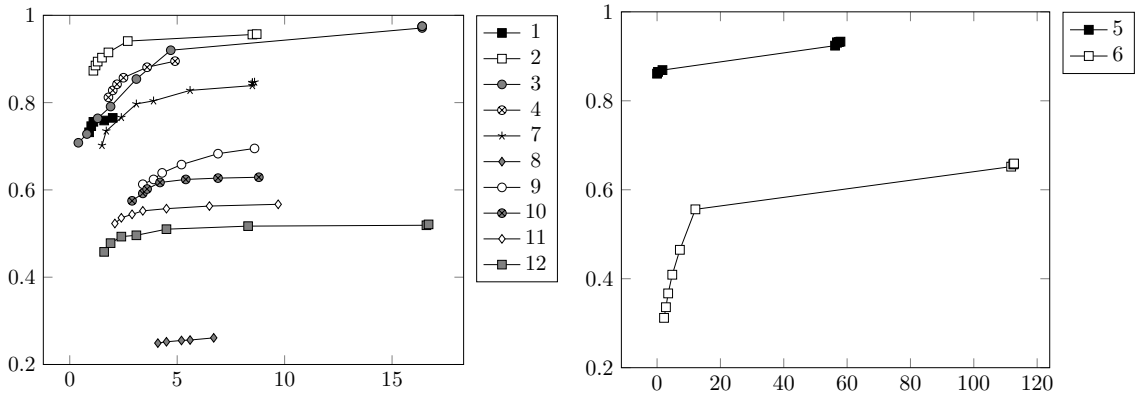

Figure 1: Average accuracy (in ordinate) and number of SUF (in abscissa) of selected results. Each curve corresponds to one data set.

A lower value of ρ therefore allows to sacrifice more of the accuracy on the training data while removing a SUF from the model. We denote by accuracy($\boldsymbol{S}$,$\mathcal{D}$) the accuracy obtained by the model $\boldsymbol{S}$ on the data $\mathcal{D}$.

Algorithm 4: Removes useless SUF from $\boldsymbol{S}$, for given $\mathcal{D}^-$ and $\rho \in [0, 1]$

```
1 end ← false
2 while end = false do
3 |   end ← true
4 |   for S ∈ S do
5 |   |   if accuracy(S\S, D⁻) ≥ ρ * accuracy(S, D⁻) then
6 |   |   |   remove S from S
7 |   |   |   end ← false
```

For each dataset and each value of $\rho \in \{0.95, 0.96, \ldots, 1.\}$, we tested this four-step process using ten-fold cross validation repeated several time. For each test, we computed the average accuracy (on validation data) and the average number of SUFs of the model. We then selected the best trade-offs between a high accuracy and a low number of SUFs involved in the model; those are represented in Figure 1. We can see that, although involving more SUFss in the model can increase accuracy, a few SUFs are often enough to reach an accuracy close to that obtained by the best max-SUF.

In order to get an idea of the accuracy that can be reached by our method, we selected the best accuracy obtained on each dataset. The 12 datasets considered in this paper have been used in [1] for evaluating the predictive accuracy of monotonic VC-DomLEM, which is a method for extracting decision rules based on the Dominance-based Rough Set Approach. Its overall accuracy on the 12 datasets is higher than those of several methods, such as SVM, OSDL [2] (instance based) or C4.5 (decision trees). Table 4 displays accuracy scores obtained by our method (with standard deviation) and those obtained by monotonic VC-DomLEM, reported from [1]. Both methods yield similar accuracy scores. To finish, Table 5 shows the distribution of rule lengths of the models with

Table 4: Accuracy of VC-DomLEM and our method on each dataset.

	1	2	3	4	5	6	7	8	9	10	11	12	avg.
Ours	76	95.3	**97.2**	89.3	92.4	65.2	84.5	26.4	**69.4**	**63**	**56.7**	53.2	72.4
±	.57	.27	.19	1.33	0.52	.34	1.48	.75	.66	.53	.71	.75	
VC-DL	**76.7**	**96.3**	97.1	**91.7**	**95.4**	**67.5**	**87.7**	**26.9**	66.7	55.6	56.4	**54.6**	72.7

Table 5: Rule length distributions, given as percentage. The number of criteria in each dataset is indicated by double vertical bars.

	Rule length															
dataset	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1		1	64	23	11	1										
2	13	72	13	1	1	1	1	1								
3			3	24	41	32										
4	38	47	10	3	1	1										
5	3	25	31	13	5	2	3	2	1	1	3	5	3	1		1
6	2	20	37	22	6	2	1	1	1	9						
7		20	21	21	12	9	15	3								
8	6	62	16	16												
9	6	26	36	32												
10	5	26	42	27												
11	6	2	25	45	9	5	4	1		3						
12	3	24	44	21	5	1				2						

the best accuracy. Rule length distribution is an interesting feature for measuring interpretability. Indeed, each decision of a classifier based on selection (or deletion) is caused by one or a few rules only. The shorter these rules, the easier it is for a user to understand the classifier decision. Again, the rule lengths distributions we obtained can be compared to those obtained by VC-DomLEM in [1]. None of the two methods provides strictly better result than the other in terms of rule length. However, a drawback of our approach is that it produces very long rules (albeit in a small percentage) on several datasets (e.g., 5, 6, 11, 12). Such long rules are less cognitively appealing.

5 Conclusion

Keeping in mind that the results of the previous section are dependent of our algorithmic choices, we can draw the following conclusions: it seems that a single Sugeno integral, even with utility functions, is not enough to give an accurate representation of monotonic datasets. In our experiments, the number of SUFs required to achieve an accuracy similar to decision rule-based models can greatly vary from one dataset to another. Moreover, Table 5 shows that rule length distributions are quite uneven, even if Step 2 tries to privilege short rules. Progress may be achieved in two directions. On the one hand we could put a restriction on the capacities, limiting ourselves to k-maxitive capacities. On the other hand, one may start by constructing a decision tree of limited depth from a dataset, and use SUFs at its leaves.

Acknowledgements. This work has been partially supported by the Labex ANR-11-LABX-0040-CIMI (Centre International de Mathématiques et d'Infor-matique) in the setting of the program ANR-11-IDEX-0002-02, subproject ISIPA (Interpolation, Sugeno Integral, Proportional Analogy).

References

[1] J. Blaszczynski, R. Slowinski, M. Szelag. Sequential covering rule induction algorithm for variable consistency rough set approaches, *Inf. Sci.*, 181: 987–1002, 2011.

[2] K. Cao-Van. *Supervised ranking from semantics to algorithms*, Ph.D. thesis, Ghent University, CS Department, 2003.

[3] M. Couceiro, D. Dubois, H. Prade, A. Rico. Enhancing the Expressive Power of Sugeno Integrals for Qualitative Data Analysis. Proc. 10th Conf. of the Europ. Soc. for Fuzzy Logic and Technology (EUSFLAT 2017), Springer, 534-547, 2017.

[4] M. Couceiro, D. Dubois, H. Prade, T. Waldhauser. Decision making with Sugeno integrals. Bridging the Gap between Multicriteria Evaluation and Decision under uncertainty. *Order*, 33 (3) 517-535, 2016.

[5] M. Couceiro, J.-L. Marichal. Characterizations of discrete Sugeno integrals as polynomial functions over distributive lattices. *Fuzzy Set. Syst.* 161 (5), 694–707, 2010.

[6] M. Couceiro, T. Waldhauser. Pseudo-polynomial functions over finite distributive lattices. *Fuzzy Set. Syst.*, 239, 21-34, 2014.

[7] D. Dubois, J.-L. Marichal, H. Prade, M. Roubens, R. Sabbadin. The use of the discrete Sugeno integral in decision making: A survey. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9:539–561, 2001.

[8] D. Dubois, H. Prade, A. Rico. The logical encoding of Sugeno integrals. *Fuzzy Sets and Systems*, 241: 61-75, 2014.

[9] M. Grabisch, T. Murofushi, M. Sugeno (Eds.) *Fuzzy Measures and Integrals. Theory and Applications*. Physica-verlag, Berlin, 2000.

[10] S. Greco, B. Matarazzo, R. Slowinski, Axiomatic characterization of a general utility function and its particular cases in terms of conjoint measurement and rough-set decision rules, *European Journal of Operational Research* 158: 271-292, 2004.

[11] J.-L. Marichal. On Sugeno integrals as an aggregation function. *Fuzzy Set. Syst.*, 114(3):347-365, 2000.

[12] M. Sugeno. Fuzzy measures and fuzzy integrals: a survey. *Fuzzy Automata and Decision Processes*, (M. M. Gupta et al., eds.), North-Holland, 89-102, 1977.