Conditioned Monotonicity for Generalized Pre-Aggregations and Aggregations

Luis Magdalena*, Daniel Gómez†, Javier Montero‡, Susana Cubillo*, Carmen Torres*

*ETSI Informáticos, Universidad Politécnica de Madrid, Spain
https://orcid.org/0000-0001-7639-8906, email:luis.magdalena@upm.es
https://orcid.org/0000-0002-6473-6039, email:susana.cubillo@upm.es
https://orcid.org/0000-0002-0340-9931, email:ctorres@fi.upm.es
†Faculty of Statistics, Complutense University of Madrid, Spain
https://orcid.org/0000-0001-9548-5781, email:dagomez@estad.ucm.es
‡Faculty of Mathematics, Complutense University of Madrid, Spain
https://orcid.org/0000-0001-8333-2155, email:monty@mat.ucm.es

Abstract—The concept of pre-aggregation function defined in [0, 1] has been recently extended to that of generalized pre-aggregation function in the framework of a totally ordered set \( T \) with maximum and minimum value. To do so, the concept of monotonicity is transformed in that of conditioned monotonicity based on the chains in \( T^n \), generalizing the idea of directional monotonicity. In the present paper we explore the concept of conditioned monotonicity considering some specific conditioning structures (covers, partitions and projections). On this basis we consider some situations where conditioned monotonicity ensures monotonicity. Finally we use these definitions and properties to define some pre-aggregation and aggregation functions that are applied to image preprocessing problems.

Index Terms—Aggregation functions, pre-aggregation functions, directional monotonicity, conditioned monotonicity.

I. INTRODUCTION

Aggregation functions [1] play a central role in information fusion and decision making. This question has been mostly focused on by means of variables taking values on [0, 1] interval, aggregated through monotonic functions. Only in recent years aggregation processes have been opened to new approaches where unit interval and monotonic functions are replaced by other options.

The [0, 1] interval can be extended to any complete lattice [2] (defining generalized aggregation functions), or even to a bounded poset as in [3] (working with type-2 fuzzy sets). A different option is that of relaxing monotonicity through weak monotonicity [4], directional monotonicity [5], or conditioned monotonicity [6]. Directional monotonicity and conditioned monotonicity are the basis for the definition of pre-aggregation functions [7] and generalized pre-aggregation functions [6].

The present paper will continue with the analysis of conditioned monotonicity and generalized pre-aggregation functions started in [6]. We will work with a totally ordered set \( T \) with maximum and minimum value. This set can even be discrete and we need to reinterpret the concept of monotonicity in terms of chains or linear orders in \( T^n \) (conditioned monotonicity). In this new framework (totally ordered sets and conditioned monotonicity), we will concentrate on some specific cases. It could be also interesting to transfer the existing relation between directional monotonicity and monotonicity, to the case of conditioned monotonicity.

The rest of the paper is structured as follows. Section II presents some preliminary concepts related to Aggregation operators, as well as the recently introduced concepts of conditioned monotonicity and generalized pre-aggregation. Then, in Section III, the concepts of conditioned monotonicity and generalized pre-aggregation are considered in the framework of partitions related to a prototype, and particularly for those partitions linked to projections. This idea leads us to analyze the monotonicity of a function in terms of conditioned monotonicity over its projections in Section IV. Then, its application to define generalized pre-aggregation and aggregation functions useful in image processing is shown in Section V. Finally, some conclusions and open questions are considered.

II. PRELIMINARIES

A. Aggregation operators

The usual definition of an aggregation operator is that of a function

\[ Ag_n : [0, 1]^n \to [0, 1] \]

which holds the following properties:

1) \( Ag_n \) is monotone, non decreasing.
2) \( Ag_n(0, \ldots, 0) = 0 \).
3) \( Ag_n(1, \ldots, 1) = 1 \).

The previous concept can be extended to a more general case by replacing the lattice [0, 1] with a complete lattice (with a maximum and minimum), leading to what is usually known as generalized aggregation function (see for example [2], [8]).

Definition 1: Let \((T, \leq)\) be a complete lattice (with maximum and minimum elements, 1 and 0 respectively) and let \((T^n, \leq_n)\) be the natural lattice of \( n \) elements of type \( T \) (i.e. \( T^n = T \times \ldots \times T \)).

A generalized aggregation function is a mapping \( Ag : T^n \to T \) such that it satisfies:

\[ Ag_n : [0, 1]^n \to [0, 1] \]
1) $Ag(0_\tau, 0_\tau, \ldots, 0_\tau) = 0_\tau$ and $Ag(1_\tau, 1_\tau, \ldots, 1_\tau) = 1_\tau$.

2) $Ag$ is monotonic with respect to the lattice’s order ($\leq n$).

But this is not the only extension that can be defined. As previously said weak monotonicity [4] extends classical monotonicity property by requiring monotonicity only along the direction of the first quadrant diagonal. This idea is further extended with the concept of directional monotonicity [5] based on that of $r$-increasing functions.

**Definition 2:** [5] Let $f$ be a function from $[0, 1]^n$ to $[0, 1]$. And let $r$ be an element of $\mathbb{R}^n$. Then we will say that $f$ is $r$-increasing if for all $x \in [0, 1]^n$ and for all $\lambda \geq 0$, the following holds:

$$f(x + \lambda r) \geq f(x).$$

Then, the concept of pre-aggregation function relies on that of $r$-increasing functions.

**Definition 3:** [7] A mapping $Ag_n : [0, 1]^n \rightarrow [0, 1]$ is an $n$-dimensional pre-aggregation function if it satisfies:

1) There exists a real vector $r \in [0, 1]^n$ with $r \neq 0$ such that $Ag_n$ is $r$-increasing.

2) $Ag_n(0, \ldots, 0) = 0$ and $Ag_n(1, \ldots, 1) = 1$.

It is also possible to merge the two previously mentioned extensions. Considering that pre-aggregation functions have already produced good results (see for example [9]) in aggregation processes working over the complete lattice $[0, 1]$, it could be interesting to extend pre-aggregations to problems defined on $T$ (a complete lattice with a maximum and minimum element).

**B. Conditioned Monotonicity**

The idea now is to extend the concept of pre-aggregation from $[0, 1]^n$ to $T^n$. To do so, the idea of directional monotonicity is adapted to a universe where directions may not exist. Consequently, it is transformed in the more general concept of conditioned monotonicity.

**Definition 4:** [6] Let $(T, \leq)$ be a totally ordered set with a maximum and minimum element $(1_\tau$ and $0_\tau$ respectively), let $(T^n, \leq n)$ be the natural lattice of $n$ elements of type $T$, let $P_\leq(T^n)$ be the set of all totally ordered subsets (chains) of $T^n$, and let $C$ be a subset of $P_\leq(T^n)$.

A mapping $F : T^n \rightarrow T$ is said to be a $C$-conditioned increasing (decreasing) function if it satisfies that $\forall a, b \in T^n$, if $a \leq b$ and $\exists C_i \in C$ such that $a, b \in C_i$, then $F(a) \leq F(b)$ ($F(a) \geq F(b)$).

In other words, the function $F$ is said to be a $C$-conditioned increasing (decreasing) function if $F|_{C_i}$ is an increasing (decreasing) function, for every chain $C_i \in C$. For this idea it is important to notice that, first, any two elements in $C_i$ are comparable since chains are linearly ordered sets, and second, any two images are comparable being the range of $F$ a subset of a totally ordered set $(T)$.

**Definition 5:** [6] Let $(T, \leq)$ be a totally ordered set with a maximum and a minimum element $(1_\tau$ and $0_\tau$ respectively), let $(T^n, \leq n)$ be the natural lattice of $n$ elements of type $T$, let $P_\leq(T^n)$ be the set of all totally ordered subsets (chains) of $T^n$, and let $C$ be a subset of $P_\leq(T^n)$.

A mapping $F : T^n \rightarrow T$ is said to be a $C$-conditioned monotonic function if it is either a $C$-conditioned increasing function or a $C$-conditioned decreasing function.

**Theorem 1:** [6] Let $(T, \leq)$ be a totally ordered set with a maximum and a minimum element $(1_\tau$ and $0_\tau$ respectively), let $(T^n, \leq n)$ be the natural lattice of $n$ elements of type $T$, let $P_\leq(T^n)$ be the set of all totally ordered subsets (chains) of $T^n$, and let $C$ be a subset of $P_\leq(T^n)$.

If $F : T^n \rightarrow T$ is a $P$-conditioned monotonic function, i.e., it is a $C$-conditioned monotonic function with $C = P_\leq(T^n)$, then $F$ is a monotonic function.

**C. Generalized pre-aggregation functions**

**Definition 6:** [6] Let $(T, \leq)$ be a totally ordered set with a maximum and a minimum element $(1_\tau$ and $0_\tau$ respectively), let $(T^n, \leq n)$ be the natural lattice of $n$ elements of type $T$, let $P_\leq(T^n)$ be the set of all totally ordered subsets (chains) of $T^n$, and let $C$ be a subset of $P_\leq(T^n)$.

A generalized pre-aggregation function over $C$ is a mapping $Ag : T^n \rightarrow T$ such that it satisfies:

1) $Ag(0_\tau, 0_\tau, \ldots, 0_\tau) = 0_\tau$ and $Ag(1_\tau, 1_\tau, \ldots, 1_\tau) = 1_\tau$.

2) $Ag$ is a $C$-conditioned increasing function.

**Proposition 1:** [6] A generalized pre-aggregation function over $P_\leq(T^n)$ is a generalized aggregation function.

**Definition 7:** [6] A mapping $Ag : T^n \rightarrow T$ is said to be a partitioning generalized pre-aggregation function over $C$ when it is a generalized pre-aggregation function over $C$, and $C$ is a partition of $T^n$.

**Proposition 2:** [6] Any pre-aggregation function in the sense of Definition 3 from [7], is a partitioning generalized pre-aggregation function.

**Definition 8:** [6] A mapping $Ag : T^n \rightarrow T$ is said to be a covering generalized pre-aggregation function over $C$ when it is a generalized pre-aggregation function over $C$, and $C$ covers $T^n$.

We say that $C$ covers $T^n$, or $C$ is a cover of $T^n$, when $C$ is a family of nonempty (and non-duplicated) subsets of $T^n$ whose union is equal to $T^n$.

**Proposition 3:** [6] Any partitioning generalized pre-aggregation function over $C$ is a covering generalized pre-aggregation function over $C$.

In summary, conditioned monotonicity offers many different options for the analysis according to the relation between the conditioning set $C$, and the overall set of chains in $T^n(P_\leq(T^n))$. These options go from monotonicity when $C = P_\leq(T^n)$ (see Proposition 1) to fusion functions (where monotonicity is eliminated) when $C = \emptyset$. 
covering conditioned monotonicity

we can replicate these same intermediate levels in conditioned
lation between conditioned monotonicity and pre-aggregation,
when $C$ is not increasing over $C$. In addition, as $F$ is a
function. In addition, as $F$ is a

$\{0, a, b, 1\}$, and the chains $C_x$ are defined by the straight line with direction $r$ that contains the element $x$.

A direction $r$ is a natural way to define a partition in $[0, 1]^n$. The question is how to define a partition in $T^n$ in a natural way similar to that of the straight line used in $r$-directional monotonicity. The main idea is to generate a partition on the basis of a certain property or condition. We will refer to this kind of partition a prototype-based partition, and to the building property or condition as the prototype of the partition.

A. Conditioned monotonicity on projective partitions

A particular and interesting case could be that of partitions where the prototype generates chains which elements are constant in every component except for one, i.e., $c \in C$ implies $c = \{a_1, \ldots, x_j, \ldots, a_n\} : x_j \in T$, and $a_i$ is constant ($i \neq j$). To explore this concept we will relate it to the idea of projection.

Definition 9: Let $T$ be a set, and let $x$ be an element of $T^n$, i.e., $x = (x_1, x_2, \ldots, x_n)$, we define the $i$-th projection map in $T^n$ as a mapping proj$_i : T^n \rightarrow T$ such that proj$_i(x) = x_i$.

We can now generalize this idea by selecting more than one component of $x$, so producing a mapping onto $T^m$ with $m \leq n$.

Definition 10: Let $T$ be a set, let $x$ be an element of $T^n$, let $J \subseteq I = \{1, 2, \ldots, n\}$, with $|J| = m$. We define the $J$-th projection map in $T^n$ as a mapping proj$_J : T^n \rightarrow T^m$ such that proj$_J(x) = (x_{j_1}, x_{j_2}, \ldots, x_{j_m})$, where $J, J_k \in J$, and $J < J_k$, when $j < k$.

On the basis of these definitions of projection we can generate equivalence relations by simply grouping in an equivalence class all those elements having the same projection.

Definition 11: Let $T$ be a set, let $x$ be an element of $T^n$, let $J \subseteq I = \{1, 2, \ldots, n\}$, let $a, b \in \Pi_J$, and let proj$_J$ be the $J$-th projection map in $T^n$. We define the equivalence relation $\Pi_J$ as $(a, b) \in \Pi_J$, if and only if proj$_J(a) = proj_J(b)$.

If we denote now the set $i$ as $i = I - \{i\}$, i.e., the complementarity of $\{i\}$ in $I$, the relation $\Pi_I$ will produce equivalence classes where all members will be identical except

Example 2.1: Let us consider the totally ordered set $(T, \leq)$ with $T = \{0, a, b, 1\}$ and the order $0 < a < b < 1$. In $(T^2, \leq^2)$, corresponding to the following Hasse diagram

covering conditioned monotonicity seems to be an interesting
option since being $C$ a cover ensures that the union of all elements in $C$ is $T^n$. Consequently, every element $x$ in $T^n$ will be in, at least, one chain ($C_x$) in $C$. This situation ensures that for any $x \in T^n$ there is at least a linear order $C_x \in C$, with $x \in C_x$, that is, every element in $T^n$ could be a matter of analysis in what concerns monotonicity. But, it could also happen the linear order not being unique.

To cope with that potential problem could be interesting to force the subsets in $C$ being pairwise disjoint. Under this restriction the cover becomes a partition of $T^n$, and the covering conditioned monotonicity is transformed into a partitioning conditioned monotonicity. The function will be $C$-conditioned monotonic, being $C$ a partition of $T^n$. In this case we can assure that for any $x \in T^n$ there exist a unique linear order $C_x \in C$, with $x \in C_x$. It is interesting to notice that the $r$-directional monotonicity defined in [5] is a particular case of partitioning conditioned monotonicity where $T = [0, 1]$, and the chains $C_x$ are defined by the straight line with direction $r$ that contains the element $x$.

The question is how to define a partition in $[0, 1]^n$. We will refer to this kind of partition a prototype-based partition, and to the building property or condition as the prototype of the partition. The main idea is to generate a partition on the basis of a certain property or condition. We will refer to this kind of partition a prototype-based partition, and to the building property or condition as the prototype of the partition. In a covering generalized pre-aggregation function, the conditioning set $C$ covers $T^n$. It makes sense to say that the conditioned monotonicity appearing in this case is a covering conditioned monotonicity. With a similar reasoning it is possible to establish the idea of partitioning conditioned monotonicity.

III. CONDITIONED MONOTONICITY ON PROTOTYPE-BASED PARTITIONS

In previous section we have considered the concept of conditioned monotonicity having its upper bound in monotonicity when $C = \mathcal{P}_\leq(T^n)$. On the other hand, applying the concept of conditioned monotonicity, we have established different levels of generalized pre-aggregations including covering and partitioning pre-aggregation functions. Having in mind the relation between conditioned monotonicity and pre-aggregation, we can replicate these same intermediate levels in conditioned monotonicity.

In a covering generalized pre-aggregation function, the conditioning set $C$ covers $T^n$. It makes sense to say that the conditioned monotonicity appearing in this case is a covering conditioned monotonicity. With a similar reasoning it is possible to establish the idea of partitioning conditioned monotonicity.
for the value of their \( i \)-th component, i.e., \((a, b) \in \Pi_T\), if and only if \( \text{proj}_i(a) = \text{proj}_i(b) \), and consequently \( a_i = b_j, \forall j \neq i \). We will represent this equivalence class as \([(a_1, a_2, \ldots, a_{i-1}, x_i, a_{i+1}, \ldots, a_n)]\). This equivalence class is made up of all elements having as common projection \((a_1, a_2, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)\).

The set of all possible equivalence classes of \( T^n \) by \( \Pi_T \), denoted \( T^n/\Pi_T \), is the quotient set of \( T^n \) by \( \Pi_T \). And any quotient set (in this case \( T^n/\Pi_T \)) is a partition of \( T^n \). We can then define a partition by means of a projection.

Let us replace now the generic set \( T \) used in previous definitions with a totally ordered set \((T, \leq)\). We will first consider the structure of its equivalence classes (being chains), and apply then the concept of conditioned monotonicity by conditioning through a projection (that generates a partition as previously shown).

**Proposition 4:** Let \((T, \leq)\) be a totally ordered set with a maximum and a minimum element \((1_T \text{ and } 0_T \text{ respectively})\), let \((T^n, \leq^n)\) be the natural lattice of \( n \) elements of type \( T \), and let \( \text{proj}_i: T^n \rightarrow T \) be the projection map over \( i \). Then, each equivalence class defined by \( \Pi_T \) is a chain in \((T^n, \leq^n)\).

**Proof:** Each equivalence class defined by \( \Pi_T \) can be represented by \([(a_1, a_2, \ldots, a_{i-1}, x_i, a_{i+1}, \ldots, a_n)]\). As \( x_i \) takes values in \( T \), if \((a_1, a_2, \ldots, a_n), (a_1, b_1, \ldots, a_n) \in [(a_1, a_2, \ldots, a_{i-1}, x_i, a_{i+1}, \ldots, a_n)]\), and \((a_1, a_2, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n) \neq (a_1, b_1, \ldots, a_{i+1}, \ldots, a_n)\), it should be either \( a_i < b_i \) or \( b_i < a_i \) (\( a_i \text{ and } b_i \) are comparable). In addition, \( \forall a \in T^n \) and \( b_1 \in T \), \( a_i < b_i \Leftrightarrow (a_1, a_2, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n) < (a_1, b_1, a_{i+1}, \ldots, a_n)\), and \( b_1 < a_i \Leftrightarrow (a_1, b_1, a_{i+1}, \ldots, a_n) < (a_1, a_2, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)\) (being \((a_1, a_2, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)\) and \((a_1, a_2, \ldots, a_{i-1}, a_{i+1}, b_1, \ldots, a_n)\) also comparable). Consequently, each pair of elements in the equivalence class defined by \( \Pi_T \) is comparable, and the equivalence class is a linear order (a chain).

If we consider again the set, and the function defined in Example 2.1, it is possible to apply the concept of projection to generate a partition and analyze conditioned monotonicity.

**Example 3.1:** Let us consider \( T = \{0, 1\} \), with the order \( 0 < a < b < 1 \), and \((T^2, \leq^2)\). We define the function \( F: T^2 \rightarrow T \) as:

\[
\begin{align*}
F(0, 0) &= F(0, a) = F(0, b) = F(a, 0) = F(b, 0) = F(b, a) = 0, \\
F(0, 1) &= F(a, a) = F(1, 0) = F(b, b) = a, \\
F(a, b) &= F(b, 1) = F(1, a) = b, \\
F(1, b) &= F(1, 1) = 1.
\end{align*}
\]

The function \( F \) is a partitioning generalized pre-aggregation function over \( T^2/\Pi_T \). The function is increasing over any chain of the form \( \{x_0, x_a, x_b, x_1\} \), with \( x \in T \), that is, over any equivalence classes of \( T^2 \).

On the other hand, \( F \) is not a partitioning generalized pre-aggregation function over \( T^2/\Pi_T \) considering that \( F \) is not increasing over the chains \( \{0a, ba, ab, 1a\} \), \( \{0b, ab, bb, 1b\} \), and \( \{01, a, b1, 11\} \).

The projections we have considered to this point produce the dimensional reduction by means of removing part of the components of \( x \). If \((T^n, \leq^n)\) has additional properties, it is possible to consider other projections when generating our partitions. We can even define a different interaction in between partitions and projections.

**Definition 12:** Let \((T, \leq)\) be a totally ordered set with a maximum and a minimum element \((1_T \text{ and } 0_T \text{ respectively})\), let \((T^n, \leq^n)\) be the natural lattice of \( n \) elements of type \( T \), and let \( x = (x_1, \ldots, x_n) \) be an element of \( T^n \). We define the lower surface of \((T^n, \leq^n)\), denoted by \( S_L(T^n) \) as:

\[ S_L(T^n) = \{ x \in T^n : \exists i \in \{1, \ldots, n\} \text{ such that } x_i = 1_T \}. \]

In a similar way we can define the upper surface of \((T^n, \leq^n)\), denoted by \( S_U(T^n) \) as:

\[ S_U(T^n) = \{ x \in T^n : \exists i \in \{1, \ldots, n\} \text{ with } x_i = 1_T \}. \]

If we define now the set \( C \subset \mathcal{P}_n(T^n) \) in such a way that it was a partition, and that every chain \( C_i \in C \) has one and only one element of \( S_L(T^n) \) (noted as \( C_{il} \)), and has one and only one element of \( S_U(T^n) \) (noted as \( C_{iu} \)), the functions \( \Pi_{C_{il}}: T^n \rightarrow S_L(T^n) \) and \( \Pi_{C_{iu}}: T^n \rightarrow S_U(T^n) \) defined as \( \Pi_{C_{il}}(x) = C_{il} \forall x \in C_i \) and \( \Pi_{C_{iu}}(x) = C_{iu} \forall x \in C_i \), respectively, are projections of \( T^n \).

This kind of partition related to a prototype/projection being not the canonical projection could also be of interest.

In particular we have said that \( r \)-directional monotonicity defined in [5] is a particular case of partitioning conditioned monotonicity where \( T = [0, 1] \), and the chains \( C_i \) are defined by the straight line with direction \( r \) that contains the element \( x \). Is it possible to define such a partition as a prototype based partition? Let us consider now this question.

In this case the definition of the partition \( C \) and the corresponding projection is quite simple.

**Definition 13:** Let us consider \( T = [0, 1] \), let \((T^n, \leq^n)\) be the natural lattice of \( n \) elements of type \( T \), given \( v = (v_1, \ldots, v_n) \in T^n, v \neq (0, \ldots, 0) \), we define the projection in the direction of \( v \) as a mapping \( \text{proj}_v: T^n \rightarrow S_L(T^n) \), such that for any \( x = (x_1, \ldots, x_n) \in T^n \), \( \text{proj}_v(x) = x - \lambda_x v \), where \( \lambda_x = \min_{v_i, v_i \neq 0} \{ x_i / v_i \} \).

But we don’t need to consider such a complicated structure if we want to create partitions in spaces as \([0, 1]^n\). We can simply create a prototype based partition on \([0, 1]^n\) by considering a prototype being a straight line with a certain slope (that at the end is also a chain). This situation corresponds to the so called directional monotonicity, being the direction that of the line (prototype). In fact, the previous idea of prototypes where only one component changes are related to this concept by taking the directions of the coordinate axes.

This idea can easily be extended to other kind of patterns or conditions (prototype chain) as a parabola or any other curve or variety, with the only restriction that the prototype should generate chains (linearly ordered sets).

**IV. FROM CONDITIONED MONOTONICITY TO MONOTONICITY**

When the conditioned monotonicity of a function appears over different partitions, it is in some cases possible to generalize it to other chains not directly considered in the partitions.
In the case of monotonicity related to different projections we can scale up the property by merging the projections. When the conditioned monotonicity appears on every projection, the function is monotonic, as will be shown below.

**Remark 1:** Let \((T, \leq)\) be a totally ordered set with a maximum and a minimum element \((1_T and 0_T\) respectively), let \((T^a, \leq^a)\) be the natural lattice of \(n\) elements of type \(T\), let \(x\) be an element of \(T^n\) and let \(\text{proj}_i : T^n \rightarrow T\) be the projection map over \(i\). If \(F : T^n \rightarrow T\) is a \(T^n/T\)-conditioned increasing function then \(\forall a \in T^n\) and \(b_i \in T, (a_1, \ldots, a_i, \ldots, a_n) \leq^a (a_1, \ldots, b_i, \ldots, a_n)\) implies that \(F(a_1, a_2, \ldots, a_n) \leq F(b_1, a_2, \ldots, a_n) \leq \ldots \leq F(b_1, b_2, \ldots, b_n)\).

**Theorem 2:** Let \((T, \leq)\) be a totally ordered set with a maximum and a minimum element \((1_T and 0_T\) respectively), let \((T^a, \leq^a)\) be the natural lattice of \(n\) elements of type \(T\), and let \(\text{proj}_i : T^n \rightarrow T\) be the projection map over \(i\). If \(F : T^n \rightarrow T\) is a \(T^n/T\)-conditioned increasing function then \(\forall i \in \{1, 2, \ldots, n\}\), then \(F\) is an increasing function in \(T^n\).

**Proof:** Let us consider \(a, b \in T^n\) such that \(a \leq b\), consequently \(a_i \leq b_i \forall i \in \{1, \ldots, n\}\). As \(F\) is \(T^n/T\)-conditioned increasing \(\forall i \in \{1, 2, \ldots, n\}\), it holds that
\[
F(a_1, a_2, \ldots, a_n) \leq F(b_1, a_2, \ldots, a_n) \leq \ldots \leq F(b_1, b_2, \ldots, b_n),
\]
where the first inequality follows from \(F\) being \(T^n/T\)-conditioned increasing, the second from \(F\) being \(T^n/T\)-conditioned increasing, and so on until the last inequality following from \(F\) being \(T^n/T\)-conditioned increasing. Then, as \(a \leq b\) implies \(F(a_1, a_2, \ldots, a_n) \leq F(b_1, b_2, \ldots, b_n)\).

\(F\) is an increasing function in \(T^n\).

**Lemma 1:** Let \((T, \leq)\) be a totally ordered set with a maximum and a minimum element \((1_T and 0_T)\) respectively, let \((T^a, \leq^a)\) be the natural lattice of \(n\) elements of type \(T\), and let \(\text{proj}_i : T^n \rightarrow T\) be the projection map over \(i\). If \(F(0_T, 0_T, \ldots, 0_T, 0_T) = 0_T, F(1_T, 1_T, \ldots, 1_T, 1_T) = 1_T\), and \(F : T^n \rightarrow T\) is a \(T^n/T\)-conditioned increasing function \(\forall i \in \{1, 2, \ldots, n\}\), then \(F\) is a generalized aggregation function in \(T^n\).

**Remark 2:** It has been previously stated ([5], [10]) that for a function \(F : [0, 1]^n \rightarrow [0, 1]\), directional monotonicity in the direction of every vector in the canonical basis, implies monotonicity. Theorem 2 could be considered as a way to adapt that result to the more general framework of the functions \(T^n \rightarrow T\).

**A. Searching for generalized aggregations**

The previous results can help us to define generalized aggregation functions in the framework of applications that work on lattices with some specific characteristics. This is the case, as an example, of several image processing problems requiring to merge the information of a tuple in \(\{0, \ldots, 255\}^n\) in a single value in \(\{0, \ldots, 255\}\). In this scenario, it is possible to find several functions that fulfill the conditions of Lemma 1 and consequently are generalized aggregation functions in \(\{0, \ldots, 255\}^n\).

Let us consider as an example, the Median of \(n\) values in \(T\).

**Lemma 2:** Let \((T, \leq)\) be a totally ordered set with a maximum and a minimum element \((1_T and 0_T)\) respectively, let \((T^a, \leq^a)\) be the natural lattice of \(n\) elements of type \(T\) with \(n\) an odd number, and let \(\phi_{\text{median}} : T^n \rightarrow T\) be the median function, \(\phi_{\text{median}}(x_1, \ldots, x_n) = \text{Median}\{x_1, \ldots, x_n\}\). Then, \(\phi_{\text{median}}\) is an increasing function in \(T^n\).

**Proof:** By Theorem 2, if \(\phi_{\text{median}} : T^n \rightarrow T\) is a \(T^n/T\)-conditioned increasing function \(\forall i \in \{1, 2, \ldots, n\}\), then \(\phi_{\text{median}}\) is an increasing function in \(T^n\).

To find the median of a set of data, the data have to be first arranged in order from least to greatest, and the median will be the middle number in the ordered set, so \(\phi_{\text{median}}(x_1, \ldots, x_n) = \phi_{\text{median}}(x_{\sigma(1)}, \ldots, x_{\sigma(n)})\), being \(\sigma : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}\) any permutation. Then,
\[
\phi_{\text{median}}(a_1, \ldots, a_{i-1}, x, a_{i+1}, \ldots, a_n) = \phi_{\text{median}}(x, a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)
\]
Thus, we can take \(i = \bar{T}\) and show that \(\phi_{\text{median}}\) is a \(T^n/T\)-conditioned increasing function.

Given \((x, a_2, \ldots, a_n) \in T^n\) with \(a_2 \leq \ldots \leq a_n\) and \(n\) an odd natural number, it holds that:
- If \(x \leq a_{n+1}\) \(\Rightarrow \phi_{\text{median}}(x, a_2, \ldots, a_n) = a_{n+1}\)
- If \(a_{n+1} < x \leq a_{n+1} + 1\) \(\Rightarrow \phi_{\text{median}}(x, a_2, \ldots, a_n) = x\), and
- If \(x > a_{n+1} + 1\) \(\Rightarrow \phi_{\text{median}}(x, a_2, \ldots, a_n) = a_{n+1} + 1\).

Let us take, \((x, a_2, \ldots, a_n) \leq (y, a_2, \ldots, a_n)\) and we can consider \(\{a_2, \ldots, a_n\}\) an ordered set. As \(x \leq y\), we have the following cases:
- If \(y \leq a_{n+1} \Rightarrow x \leq y \leq a_{n+1}\)
  - and \(\phi_{\text{median}}(y, a_2, \ldots, a_n) = a_{n+1}\)
  - \(\phi_{\text{median}}(y, a_2, \ldots, a_n) = \phi_{\text{median}}(a_2, \ldots, a_n)\),
- If \(x \leq a_{n+1} \leq y \leq a_{n+1} + 1\) \(\Rightarrow \phi_{\text{median}}(x, a_2, \ldots, a_n) = a_{n+1}\),
  - \(\phi_{\text{median}}(y, a_2, \ldots, a_n) = y\) and \(a_{n+1} < y\),
- If \(x \leq a_{n+1} < a_{n+1} + 1 < y \Rightarrow \phi_{\text{median}}(x, a_2, \ldots, a_n) = a_{n+1}\),
  - \(\phi_{\text{median}}(y, a_2, \ldots, a_n) = a_{n+1} + 1\), and \(a_{n+1} < a_{n+1} + 1\) since \(\{a_2, \ldots, a_n\}\) is an ordered set.
- If \(a_{n+1} + 1 < x \leq y \Rightarrow \phi_{\text{median}}(x, a_2, \ldots, a_n) = \phi_{\text{median}}(y, a_2, \ldots, a_n) = a_{n+1} + 1\), and \(x \leq a_{n+1} + 1\).
- If \(x \leq a_{n+1} + 1 < y \Rightarrow \phi_{\text{median}}(x, a_2, \ldots, a_n) = \phi_{\text{median}}(y, a_2, \ldots, a_n) = a_{n+1} + 1\), and \(x \leq a_{n+1} + 1\).

Therefore, \(\phi_{\text{median}}\) is a \(T^n/T\)-conditioned increasing function.

**Remark 3:** The demonstration can be extended to the case of \(n\) being an even number by defining the median as either:
- \(\phi_{\text{median}}(x_1, \ldots, x_n) = x_{\sigma(\lfloor \frac{n}{2} \rfloor)}\) or \(\phi_{\text{median}}(x_1, \ldots, x_n) = x_{\sigma(\lceil \frac{n}{2} \rceil)}\), being \(\sigma\) an ordering permutation such that \(x_{\sigma(1)} \leq x_{\sigma(2)} \leq \ldots \leq x_{\sigma(n)}\).
Proposition 5: The function $\phi_{\text{median}}$ is a generalized aggregation function for $T = \{0, \ldots, 255\}$.

Proof: Following Lemma 2, $\phi_{\text{median}}$ is a monotonic non-decreasing function. In addition, $\phi_{\text{median}}(0, 0, \ldots, 0) = 0$ and $\phi_{\text{median}}(255, 255, \ldots, 255) = 255$. Consequently, $\phi_{\text{median}} : \{0, \ldots, 255\}^n \rightarrow \{0, \ldots, 255\}$ is a generalized aggregation function.

It is important to emphasize that in the same way that the concept of directional monotonicity for Riesz spaces defined in [5] allows to extend the set of aggregation operators that can be used for soft computing, the directional monotonicity concept here introduced extends in a more general way the set of pre-aggregation functions allowing now to deal with any space $T$ in which a partial order is defined.

Let us now present an aggregation operator $T$ that can be viewed as a Generalized Pre-Aggregation but is not a pre-aggregation in the sense of [5] nor a generalized aggregation operator.

Example 5.1: The first example that we present in this section is a classical smoothed aggregation of a grey-scale image. Let us assume that we have a grey-scale image $I_{\text{gray}} = I_{i,j} \in \{0, 1, \ldots, 255\}$.

Now let us define the neighborhood $N(i, j)$ of each pixel with coordinates $(i, j)$. The idea of the smoothing process is to define a new image that is the result of aggregate to each pixel the spectral information of its neighborhood $N(i, j)$. Frequently, with the intention of softening the original image $I_{\text{gray}}$, another $I^*_{\text{gray}} = \{I^*(i, j) \mid i = 1, \ldots, s; j = 1, \ldots, r\}$ where

$$I^*(i, j) = \phi(I(u, v), (u, v) \in N(i, j))$$

with

$$\phi : [0, 255]^n \rightarrow [0, 255]$$

being an aggregation function in the classical sense.

Given a pixel $P_{i,j}$, let us suppose that $N(i, j) = \{(i-1,j-1), (i-1,j), (i-1,j+1), (i,j-1), (i,j), (i,j+1), (i+1,j-1), (i+1,j), (i+1,j+1)\}$. If we denote by $P_{N(i,j)}$ the 9 dimensional vector

$$P_{N(i,j)} = (P_{i-1,j-1}, P_{i-1,j}, P_{i-1,j+1}, P_{i,j-1}, P_{i,j}, P_{i,j+1}, P_{i+1,j-1}, P_{i+1,j}, P_{i+1,j+1})$$

a smoothed aggregation operator could be defined in a natural way as:

$$\phi_{\text{smoo}}: \{0, \ldots, 255\}^9 \rightarrow \{0, \ldots, 255\}_{P_{N(i,j)}} \text{ Median}\{P_{(u,v)} : (u,v) \in N(i,j)\}$$

With this definition, as shown in previous section, $\phi_{\text{smoo}}$ is a Generalized aggregation function.

The smoothing methods of an image are especially useful in edge detection problems. Just to test the effectiveness of the smoothing aggregation here presented, the edge detection algorithm known as Sobel has been applied after smoothing the image using the $\phi_{\text{smoo}}$ aggregator and compared with the output provided directly by the Sobel algorithm. The following table shows the results obtained for the first 50 known images of the Berkeley [14] database with the maximum human pairing method.

As it can be seen, the results obtained after the application of $\phi_{\text{smoo}}$ significantly improve the overall performance, since they decrease the average recall but increasing the precision and the $F$ measured.

Example 5.2: Let us now present an aggregation operator that can be viewed as a Generalized Pre-Aggregation but is neither a pre-aggregation in the sense of [5] nor a generalized aggregation operator. In edge detection problems it is frequent to identify changes in luminosity in different directions. To do so it is common to transform the original image $I_{\text{gray}}$ into

$$\tilde{I} = \phi_{\text{median}}(I_{\text{gray}}; (i,j) \in N(i,j))$$

Nevertheless, most of aggregation operators used in this topic are defined on the real interval $[0, 255]$ or its normalized version $[0, 1]$. Let us analyze the following two very well-known examples in which aggregation operators should be defined over a discrete domain.

As it can be deduced from the previous formulation, we can conclude that when working with grey-scale images or color images, the associated domain for each pixel with coordinates $(i, j)$ for a channel $k$ (that is, $P_{i,j}^k$) should be the discrete set $T = \{0, \ldots, 255\}$. Taking into account this fact, it would be natural to think that any aggregation operator that deals with this class of spectral information should have the same set.
another that gathers the differences in some specific directions (see for example [15]–[17]).

Given a pixel $P_{i,j}$, with intensity $I_{\text{gray}}(i,j)$, let us denote by $I_{\text{gray}}(i,j)^0 \in \{0, \ldots, 255\}$ the gray scale value obtained after applying the function $\phi$ to the intensity of the pixel $P_{i,j}$. And let us denote by $I_{\text{gray}}$ the associated image. In [17], two functions $\phi_v$ and $\phi_h$ were defined to detect directional changes in luminosity. Formally, given a pixel with coordinates $(i, j)$ we have:

- $\phi_v : \{0, \ldots, 255\}^0 \rightarrow \{0, \ldots, 255\}$

- $\phi_h : \{0, \ldots, 255\}^0 \rightarrow \{0, \ldots, 255\}$

\[
\phi_v(I_{\text{gray}}(i-1, j-1), \ldots, I_{\text{gray}}(i+1, j+1)) = \max\{0, I_{\text{gray}}(i, j+1) - I_{\text{gray}}(i, j-1)\}.
\]

\[
\phi_h(I_{\text{gray}}(i-1, j-1), \ldots, I_{\text{gray}}(i+1, j+1)) = \max\{0, I_{\text{gray}}(i, j+1) - I_{\text{gray}}(i, j-1)\}.
\]

Figure 1 illustrates these two operators detecting directional changes. $\phi_v$ corresponds to the changes in the direction described as $V(1,0)$ while $\phi_h$ corresponds to $V(0,1)$.

It is very easy to check that the functions $\phi_v$ and $\phi_h$ are not classical generalized aggregation operators since the monotonicity condition is not satisfied.

For example if we take the inputs $x = (0, 0, 0, 70, 0, 120, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$, $y = (0, 0, 0, 80, 0, 120, 0, 0, 0, 0) \in \{0, \ldots, 255\}^9$. Its clear that $x \leq y$ in the partial order of the set $\{0, \ldots, 255\}$, but $\phi_v(x) = 120 - 70 > \phi_v(y) = 120 - 80 = 40$, and $\phi_h(x)$ is not a monotonic increasing function. Also it is easy to check that these functions are not pre-aggregation in the sense of [5] since we have a discrete space.

Nevertheless, we can see that there exit directions (the sixth coordinate i.e. $r = (0, 0, 0, 0, 0, 1, 0, 0, 0)$) in which the $\phi_h$ is monotone.

So it is not very difficult to see that the functions $\phi_v$ and $\phi_h$ are Generalized Pre-Aggregations.

The Generalized Pre-Aggregation function $\phi_v$ is growing in the sixth coordinate since we are searching the changes from right to left in the image (see Figure 2 and Figure 3) and we want that the function increases in the position $(i, j+1)$ that corresponds with the sixth coordinate.

### VI. CONCLUSIONS

The present paper explores the concept of conditioned monotonicity considering some specific conditioning structures (covers, partitions and projections). On this basis we study some situations where conditioned monotonicity ensures monotonicity. Finally we use these definitions and properties to consider some generalized pre-aggregation and aggregation functions that correspond to operators commonly applied to image preprocessing problems. In this way we include under the umbrella of generalized pre-aggregation functions those operators that were not previously considered as aggregations.
Fig. 2. Original, vertical and horizontal images after applying the Generalized Pre-Agregation $\phi_v$ and $\phi_h$ ($I_{126007}^\phi$ image from Berkeley dataset).

ACKNOWLEDGMENT
This research has been partially supported by the Government of Spain (grant PGC2018-096509-B-I00), the Government of Madrid (grant S2013/ICCE-2845), Complutense University (UCM Research Group 910149 and grant PR26/16-21B-3) and Universidad Politécnica de Madrid.

REFERENCES