Majority Additive–Ordered Weighting Averaging: A New Neat Ordered Weighting Averaging Operator Based on the Majority Process

J. I. Peláez*, J. M. Doña
Department of Languages and Computer Sciences, University of Malaga, Malaga 29071, Spain

A problem that we had encountered in the aggregation process is how to aggregate the elements that have cardinality greater than one. The most common operators used in the aggregation process produce reasonable results, but, at the same time, when the items to aggregate have cardinality greater than one, they may produce distributed problems. The purpose of this article is to present a new neat ordered weighting averaging (OWA) operator that uses the cardinality of these elements to calculate their weights. © 2003 Wiley Periodicals, Inc.

1. INTRODUCTION

The process of information aggregation appears, in many applications, to be related to the development of intelligent systems. Ones sees aggregation in neural networks, fuzzy logic controllers, vision systems, expert systems, and multicriteria decision aids. Yager[1] introduced a new aggregation technique based on the ordered weighted averaging (OWA) operators. These OWA operators can provide for aggregations lying between the logical *or* and *and.* Yager[1] defined an OWA operator of dimension n as a mapping

$$F : R^n \rightarrow R$$

that has an associated n vector $\mathbf{W} = [w_1, w_2, \ldots, w_n]^T$ such that $w_i \in [0, 1]$ and $\sum_{i=1}^{n} w_i = 1$. Furthermore, $F(a_1, a_2, \ldots, a_n) = \sum_{j=1}^{n} w_j \cdot b_j$, where b_j is the jth largest of the a_i.

Example. Assume $\mathbf{W} = [0.4, 0.3, 0.2, 0.1]^T$, then

*Author to whom all correspondence should be addressed: e-mail: jignacio@lcc.uma.es.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 18, 469–481 (2003)
 Published online in Wiley InterScience (www.interscience.wiley.com). • DOI 10.1002/int.10096

$$F(0.7, 1, 0.3, 0.6) = (0.4)(1) + (0.3)(0.7) + (0.2)(0.6) + (0.1)(0.3) = 0.76$$

A fundamental aspect of this operator is the reordering step; in particular, an aggregate a_i is not associated with a particular weight w_i, but rather a weight is associated with a particular ordered position of aggregate.

When we view the OWA weights as a column vector, we shall find it convenient to refer to the weights with the low indices as weight at the top and those with the higher indices with weights at the bottom.

It is noted that different OWA operators are distinguished by their weighting function. Yager[1] pointed out three important special cases of OWA aggregations:

(1) F^*—in this case $\mathbf{W} = \mathbf{W}^* = [1, 0, \ldots, 0]^T$
(2) F_*—in this case $\mathbf{W} = \mathbf{W}_* = [0, 0, \ldots, 1]^T$
(3) F_{ave}—in this case $\mathbf{W} = \mathbf{W}_{\text{ave}} = [1/n, 1/n, \ldots, 1/n]^T$

It can easily be seen that

(1) $F^*(a_1, a_2, \ldots, a_n) = \max(a_i)$
(2) $F_*(a_1, a_2, \ldots, a_n) = \min(a_i)$
(3) $F_{\text{ave}}(a_1, a_2, \ldots, a_n) = (1/n) \cdot \sum_{i=1}^{n} a_i$

A number of important properties can be associated with these operators.[1–5] Now, we shall discuss some of these.

PROPERTY 1. *For any OWA operator F*

$$F_*(a_1, a_2, \ldots, a_n) \leqslant F(a_1, a_2, \ldots, a_n) \leqslant F^*(a_1, a_2, \ldots, a_n)$$

Thus, the upper and lower star OWA operator are boundaries. From the foregoing equations, it becomes clear that for any F

$$\min(a_i) \leqslant F(a_1, a_2, \ldots, a_n) \leqslant \max(a_i)$$

PROPERTY 2. *Commutative: The OWA operator can be seen as commutative. Let* $\langle a_1, a_2, \ldots, a_n\rangle$ *be a bag aggregate and let* $\langle d_1, d_2, \ldots, d_n\rangle$ *be a permutation of the* a_i. *Then, for any OWA operator*

$$F(a_1, a_2, \ldots, a_n) = F(d_1, d_2, \ldots, d_n)$$

PROPERTY 3. *Monotonicity: A third characteristic associated with the operators is monotonicity. Assume* a_i *and* c_i *are a collection of aggregates* $i = 1, \ldots, n$ *such that for each* i, $c_i \leqslant a_i$. *Then,*

$$F(c_1, c_2, \ldots, c_n) \leqslant F(a_1, a_2, \ldots, a_n)$$

where F is some fixed weight OWA operator.

PROPERTY 4. *Idempotency: Another characteristic associated with these operators is idempotency. If* $a_i = a$ *for all* $i = 1, \ldots, n$, *then for any OWA operator*

$$F(a, a, \ldots, a) = a$$

From the foregoing equation we can see the OWA operators have the basic properties associated with an averaging operator.

In addition, Yager[1] defined two important measures associated with an OWA operator. The first measure, called the dispersion (or entropy), of an OWA vector **W** is defined as

$$\mathrm{Disp}(\mathbf{W}) = -\sum_{i=1}^{n} w_i \cdot \ln(w_i)$$

When using the OWA operator as an averaging operator, we can see that Disp(**W**) measures the degree to which we use all the aggregates equally.

The second measure, which we shall call the *or-ness* measure is defined as

$$\text{or-ness}(\mathbf{W}) = \frac{1}{n-1} \sum_{i=1}^{n} ((n-i) \cdot w_i)$$

It can easy be shown that

(1) or-ness($\mathbf{W}^*$) = 1
(2) or-ness($\mathbf{W}_*$) = 0
(3) or-ness($\mathbf{W}_{ave}$) = 0.5

A measure of *and-ness* can be defined as: and-ness(**W**) = 1 − or-ness(**W**)

Generally, it should be appreciated that an OWA operator with much of the weights near the top will be an *or-like* operator or-ness(**W**) ⩾ 0.5. At the other extreme, when the weights are nonzero near the bottom, the OWA operator will be *and-like* or-ness(**W**) ⩽ 0.5.

The OWA operators have been applied in different areas.[6] In this way, for example, Yager[7] described their use in modeling natural networks in Ref. 1 applies it to multicriteria decision making; in Ref. 8 Yager discussed its use in database systems, in Ref. 9 use in fuzzy systems modeling, in Ref. 10 use in multicriteria aggregation problems where we need only to satisfy some portion of the criteria, in Ref. 5 information fusion, and in Ref. 11 use it in modeling nearest neighbor rules. O'Hagan[12] has suggested their use in expert systems; Yager and Filev[13] have used these operators to develop flexible models for fuzzy logic controllers; Marimin et al.[14] apply it to fuzzy group decision making; Mitchell and Shaefer[15] use it in predictive picture compression and multiple-attribute classification; Torra[16] shows how it can be extended to deal with linguistic labels; Herrera et al.[17] use it for group decision making using linguistic OWA Operators.

The most common operators used in the aggregation[18] produce reasonable results, but at the same time, they may produce distributed problems. This problem is a variation of the cake-cutting problems,[19] where we need to divide a cake in a way that all fellow guests were satisfied with their portions. To solve this type of problem, we propose a new method for aggregation denominated *majority,* which uses the cardinality of the elements to aggregate.

In this work, we continue with the investigation of the OWA operators and we present a new neat OWA operator. This operator is based on the majority process and we denominated it as the *majority additive*–OWA (MA-OWA) operator. This study is organized as follows: in Section 2 we review the neat OWA operators. In Section 3, we discuss the majority process and, finally, in Section 4, we present the MA-OWA operator.

2. NEAT OWA OPERATORS

In the OWA operators we have assumed that the weights were fixed given constant values. In this section we shall generalize the concept of OWA aggregation by allowing the weights to be a function of the aggregates, more precisely, the ordered aggregates b_i. It is still required that the weights satisfy normalization conditions. In this case

$$F(a_1, a_2, \ldots, a_n) = \sum_{i=1}^{n} w_i \cdot b_i$$

where $w_i = f_i(b_1, b_2, \ldots, b_n)$.

In this case, where the weights depend on the aggregates, many, but not all, of the properties of the OWA operator still hold:

(1) All aggregate-dependent operators still lie between F^* and F_*.
(2) The operator is still idempotent

$$F(a, a, \ldots, a) = a$$

(3) The operator is still commutative; the indexing of the a_i's is not important.

One property that is not necessarily retained is that of monotonicity. Assume $A = (a_1, a_2, \ldots, a_n)$ and

$$\hat{A}' = (\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_n)$$

are two collections aggregates such that $a_i \geqslant \hat{a}_i$ for all i. If the weights are constant, then

$$F(A) \geqslant \hat{F}(\hat{A})$$

Subsequently, we shall see that this is not the case when the weights are aggregate dependent.

We shall say that an OWA aggregation is neat if the aggregated value is independent of the ordering. Let $A = (a_1, a_2, \ldots, a_n)$ be our inputs ordered, $B = (b_1, b_2, \ldots, b_n)$ be our inputs ordered, and $C = (c_1, c_2, \ldots, c_n) = \text{Perm}(a_1, a_2, \ldots, a_n)$ be a permutation of the input. Formally, we shall say that the OWA operator is neat if

$$F(a_1, a_2, \ldots, a_n) = \sum_{i=1}^{n} w_i \cdot b_i$$

is the same for any assignment $C = B$.

One simple example of a neat OWA operator is when $w_i = 1/n$. In this case,

$$F(a_1, a_2, \ldots, a_n) = \frac{1}{n} \cdot \sum_{i=1}^{n} a_i$$

If the weight w_j's are fixed, this is the one possible neat OWA operator. However, as we shall see, when we allow aggregate-dependent weights, we can get more neat operators.

One important characteristic of neat OWA aggregators is that they don't need to be ordered. This implies that the formulation for the neat OWA aggregators can be written using the arguments a_i's, by introducing the ordered inputs b_i's.

There are several families of neat operators.[18] A first family of aggregate-dependent weight that we shall study is the so-called BADD OWA operators.[18] In the case of these operators, we define

$$w_i = \frac{b_i^\alpha}{\sum_{j=1}^{n} b_j^\alpha}$$

It is easy to see that it completes the normalization conditions.

Thus, these are an acceptable class of OWA weights. In this case we get

$$F(a_1, a_2, \ldots, a_n) = \frac{\sum_{i=1}^{n} b_i^{\alpha+1}}{\sum_{i=1}^{n} b_i^\alpha}$$

It is easy to see that this is a neat OWA operator and, thus,

$$F(a_1, a_2, \ldots, a_n) = \frac{\sum_{i=1}^{n} a_i^{\alpha+1}}{\sum_{i=1}^{n} a_i^\alpha}, \quad \alpha \geqslant 0$$

Hence, no ordering is necessary in this process. We see for the case when $\propto = 0$, we get

$$F(a_1, a_2, \ldots, a_n) = \frac{1}{n} \cdot \sum_{i=1}^{n} a_i$$

which is the simple average F_{ave}. When $\propto = 1$ we get

$$F(a_1, a_2, \ldots, a_n) = \frac{\sum_{i=1}^{n} a_2^2}{\sum_{j=1}^{n} a_j}$$

When $\propto \to \infty$ we have

$$F(a_1, a_2, \ldots, a_n) = \max[a_i] = F^*(a_1, a_2, \ldots, a_n)$$

It can be seen that for this class of operators, $w_i \geqslant w_i$ for $i > j$; thus, it follows because $b_i \geqslant b_j$ for $i < j$. Therefore, these are buoyancy operators and as such are or-like operators.

In another class of aggregate-dependent weights, we let

$$w_i = \frac{\left(\frac{1}{b_i}\right)^{\alpha}}{\sum_{j=1}^{n} \left(\frac{1}{b_j}\right)^{\alpha}}$$

These, again, can be shown to satisfy the conditions of OWA weights. In this case

$$F(a_1, a_2, \ldots, a_n) = \frac{\sum_{i=1}^{n} \left(\frac{1}{b_i}\right)^{\alpha-1}}{\sum_{i=1}^{n} \left(\frac{1}{b_i}\right)^{\alpha}} = \frac{\sum_{i=1}^{n} \left(\frac{1}{a_i}\right)^{\alpha-1}}{\sum_{i=1}^{n} \left(\frac{1}{a_i}\right)^{\alpha}}$$

Thus, these also are neat. We see that if $\propto = 0$, we get

$$F(a_1, a_2, \ldots, a_n) = \frac{\sum_{i=1}^{n} \left(\frac{1}{a_i}\right)^{-1}}{\sum_{i=1}^{n} 1} = \frac{\sum_{i=1}^{n} a_i}{n} = \frac{1}{n} \cdot \sum_{i=1}^{n} a_i$$

Thus, when $\propto = 0$, we get F_{ave}. If $\propto = 1$, we get

$$F(a_1, a_2, \ldots, a_n) = \frac{\sum_{i=1}^{n} 1}{\sum_{i=1}^{n} \frac{1}{a_i}} = \frac{n}{\sum_{i=1}^{n} \frac{1}{a_i}}$$

which is the Harmonic mean. If we let $\propto \to \infty$, we then get

$$F(a_1, a_2, \ldots, a_n) = a_{\min} = F$$

Another interesting case of aggregate-dependent weights is where

$$w_i = \frac{(1 - b_i)^{\alpha}}{\sum_{i=1}^{n} (1 - b_i)^{\alpha}}$$

In this case we get

$$F(a_1, a_2, \ldots, a_n) = \frac{\sum_{i=1}^{n} (1 - a_i)^{\alpha} \cdot a_i}{\sum_{i=1}^{n} (1 - a_i)^{\alpha}}$$

which, again, is a neat aggregation. When $\propto = 0$ we get F_{ave}.

$$F(a_1, a_2, \ldots, a_n) = \frac{1}{n} \cdot \sum_{i=1}^{n} a_i = F_{\text{ave}}$$

When $\propto = 1$, we get

$$F(a_1, a_2, \ldots, a_n) = \frac{\sum_{i=1} (a_i - a_i^2)}{n - \sum_{i=1}^{n} a_i}$$

When $\propto = \infty$, we get

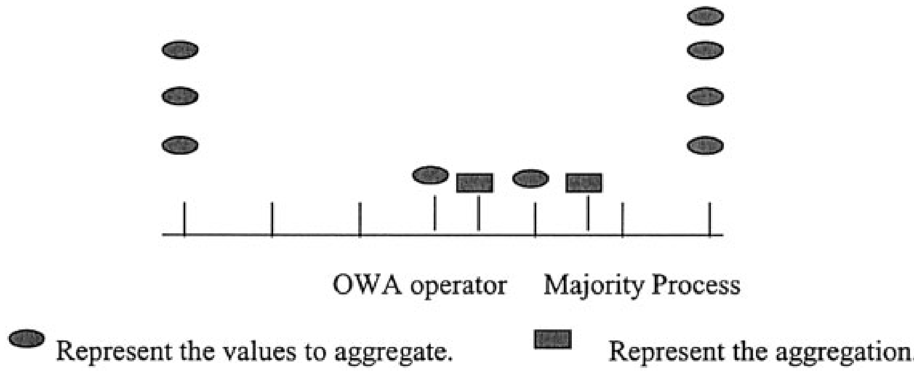

Figure 1. Graphics representation using an OWA operator (in this case $w_i = 1/n$) and majority operator. The result produced by other operators may produce distributed problems.

$$F(a_1, a_2, \ldots, a_n) = \min_{i=1,\ldots,n} a_i$$

3. MAJORITY PROCESS

Majority processes arise because the most common aggregation operators[18] overemphasize the opinion of the minority at the expense of those of the majority, creating an aggregation that may be considered imprecise for the majority. We call these distribution problems a variation of the cake-cutting problems.[19] For example, if we want to aggregate the elements {0.7, 0.7, 0.7, 0.7, 0.5, 0.4, 0.1, 0.1, 0.1}, the classical operators consider nine sets with cardinality 1; nevertheless, the majority process considers four sets with cardinality 4, 1, 1, and 3, respectively. If we analyze this example (Figure 1), 55% of the elements over 0.4 and 44% are the elements 0.7. The result should be higher than 0.5 and lower than 0.7 because we obtain a value that represents the majority and at the same time, the minority. Using the majority process, we obtain this type of result (0.551), and we eliminate the distribution problems that other operators may produce.

These distribution problems result from the process through which the items are considered. Classical aggregation considers items in an independent manner and majority processes consider groups of items with different cardinalities. Figure 2 graphically represents this difference. This example has six elements to aggregate. Classical aggregation processes all these elements independently whereas the majority aggregation first groups items by their similarities and then aggregates. Depending on the manner in which items are aggregated (Figure 2), majority processes emerge.

Majority aggregation processes are completed as follows: (1) select an element from each group and aggregate the elements; (2) subtract 1 from the cardinality of each group and eliminate those groups with a cardinality of 0; (3) with the results of the aggregation in the first two steps, create a new group with

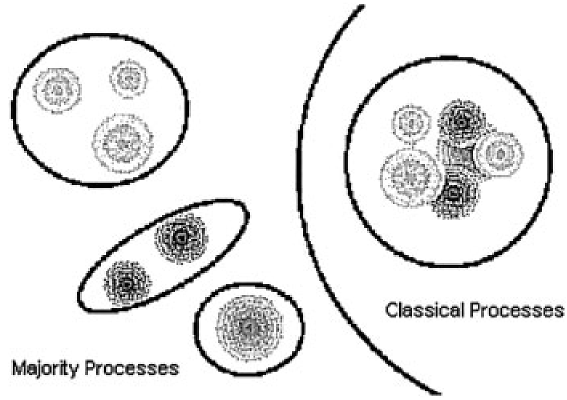

Figure 2. Representation of the majority and classical aggregation.

a cardinality of 1; and (4) repeat the previous steps until only one group remains. Figure 3 graphically shows this majority aggregation process.

The results obtain an aggregation value that represents the majority but at the same time introduces in larger or smaller ways the preferences of the minority and in this way creates a more precise aggregation that avoids distribution problems.

4. MA-OWA OPERATORS: A NEAT OPERATOR

The MA-OWA operator is a modification of the arithmetic mean, as an arithmetic mean of arithmetic mean, such that, the final result is a weighted arithmetic mean.

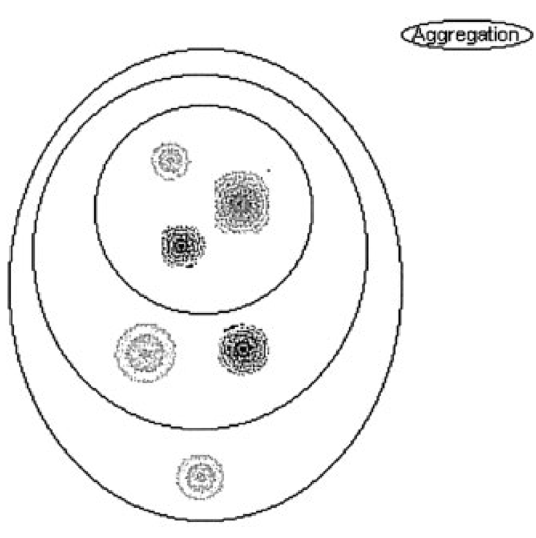

Figure 3. Majority aggregation.

DEFINITION 1. *We define the MA-OWA as*

$$F_{\mathrm{MA}}(a_1, a_2, \ldots, a_n) = \sum_{j=1}^{n} w_j \cdot b_j = \sum_{j=1}^{n} f_j \cdot (b_1, b_2, \ldots, b_n) \cdot b_j$$

where $w_j \in [0, 1]$ *and* $\sum_{j=1}^{n} w_j = 1$.

Furthermore, $F_{\mathrm{MA}}(a_1, a_2, \ldots, a_n) = \sum_{j=1}^{n} w_j \cdot b_j$, where b_j is the jth largest of the a_i and

$$w_j = f_j(b_1, \ldots, b_n) = \frac{1}{\prod_{k=g_j}^{n} h_k(b_1, \ldots, b_n)}$$

where g_j is a function that indicate when the b_j element is used in the aggregation process. The h_k is a function that indicates the number of elements in each step in the aggregation process.

This operator is a neat OWA. Their weights are calculated in functions of the cardinality of the elements to aggregate and they are independent of the order.

Now, we are going to show that as the weights satisfy the normalization conditions,

(1) $\sum_{i=1}^{n} w_i = 1$. To show that the weights sum is equal to one, we consider an equivalent equation with the previous one, where we sum the equal elements' weights. In this case, the weights of MA-OWA operators may be calculated as follows. Let $\boldsymbol{\delta i}$ be the cardinality for the element $i/\boldsymbol{\delta i} > 0$, then,

$$w_i = \frac{\gamma_i^{\delta_{\min}}}{\theta_{\delta_{\max}} \cdot \theta_{\delta_{\max+1}} \cdot \cdots \cdot \theta_{\delta_{\min+1}} \cdot \theta_{\delta_{\min}}} + \frac{\gamma_i^{\delta_{\min+1}}}{\theta_{\delta_{\max}} \cdot \theta_{\delta_{\max+1}} \cdot \cdots \cdot \theta_{\delta_{\min+1}}} + \cdots + \frac{\gamma_i^{\delta_{\max-1}}}{\theta_{\delta_{\max}} \cdot \theta_{\delta_{\max+1}}} + \frac{\gamma_i^{\delta_{\max}}}{\theta_{\delta_{\max}}}$$

where

$$\gamma_i^k = \begin{cases} 1 & \text{if } \delta_i \geqslant k \\ 0 & \text{otherwise} \end{cases}$$

and

$$\theta_i = \begin{cases} (\text{number of items with cardinality} \geqslant i) + 1 & \text{if } i \neq \boldsymbol{\delta}_{\min} \\ \text{number of items with cardinality} \geqslant i & \text{otherwise} \end{cases}$$

If the first term of all weighted w is added

$$\sum_{i=1}^{n} \frac{\gamma_i^{\delta_{\min}}}{\theta_{\delta_{\max}} \cdot \theta_{\delta_{\max+1}} \cdot \cdots \cdot \theta_{\delta_{\min+1}} \cdot \theta_{\delta_{\min}}} = \frac{\theta_{\delta_{\min}}}{\theta_{\delta_{\max}} \cdot \theta_{\delta_{\max+1}} \cdot \cdots \cdot \theta_{\delta_{\min+1}} \cdot \theta_{\delta_{\min}}} = \frac{1}{\theta_{\delta_{\max}} \cdot \theta_{\delta_{\max+1}} \cdot \cdots \cdot \theta_{\delta_{\min+1}}}$$

and the second term is added also

$$\sum_{i=1}^{n} \frac{\gamma_i^{\delta_{\min+1}}}{\theta_{\delta_{\max}} \cdot \theta_{\delta_{\max+1}} \cdot \cdots \cdot \theta_{\delta_{\min+1}}} = \frac{\theta_{\delta_{\min+1}} - 1}{\theta_{\delta_{\max}} \cdot \theta_{\delta_{\max+1}} \cdot \cdots \cdot \theta_{\delta_{\min+1}}}$$

adding these values it obtained

$$\frac{1}{\theta_{\delta_{\max}} \cdot \theta_{\delta_{\max+1}} \cdot \cdots \cdot \theta_{\delta_{\min+1}}} + \frac{\theta_{\delta_{\min+1}} - 1}{\theta_{\delta_{\max}} \cdot \theta_{\delta_{\max+1}} \cdot \cdots \cdot \theta_{\delta_{\min+1}}} = \frac{1}{\theta_{\delta_{\max}} \cdot \theta_{\delta_{\max+1}} \cdot \cdots \cdot \theta_{\delta_{\min+2}}}$$

This process is repeated for the rest of the terms. For the last elements, it obtained

$$\sum_{i=1}^{n} \frac{\gamma_i^{\delta_{\max-1}}}{\theta_{\delta_{\max}} \cdot \theta_{\delta_{\max+1}}} = \frac{\theta_{\delta_{\max+1}} - 1}{\theta_{\delta_{\max}} \cdot \theta_{\delta_{\max+1}}}$$

with the previous values

$$\frac{\theta_{\delta_{\max+1}} - 1}{\theta_{\delta_{\max}} \cdot \theta_{\delta_{\max+1}}} + \frac{1}{\theta_{\delta_{\max}} \cdot \theta_{\delta_{\max+1}}} = \frac{1}{\theta_{\delta_{\max}}}$$

and, finally,

$$\sum_{i=1}^{n} \frac{\gamma_i^{\delta_{\max}}}{\theta_{\delta_{\max}}} = \frac{\theta_{\delta_{\max}} - 1}{\theta_{\delta_{\max}}}$$

when all is added

$$\frac{\theta_{\delta_{\max}} - 1}{\theta_{\delta_{\max}}} + \frac{1}{\theta_{\delta_{\max}}} = 1$$

(2) $w_i \in [0, 1]$.

Proof 2. Trivial with the display in 1.

We point out three important cases for this operator:

(1) If $\delta_i = 1$ for all $i = 1, \ldots, n$, $F_{\mathrm{MA}}(a_1, a_2, \ldots, a_n) = 1/n \sum_i a_i = F_{\mathrm{ave}}$
(2) If $\delta_n \to \infty$, $F_{\mathrm{MA}}(a_1, a_2, \ldots, a_n) = \max[b_i] = F^*(b_1, b_2, \ldots, b_n)$
(3) If $\delta_1 \to \infty$, $F_{\mathrm{MA}}(a_1, a_2, \ldots, a_n) = \min[b_i] = F_*(b_1, b_2, \ldots, b_n)$

4.1. Properties

In the following, we present the principal properties for the MA-OWA operator.

PROPERTY 1. *The MA-OWA is an or/and operator*

$$F_*(a_1, a_2, \ldots, a_n) \leqslant F_{\mathrm{MA}}(a_1, a_2, \ldots, a_n) \leqslant F^*(a_1, a_2, \ldots, a_n)$$

Thus, the superscript and subscript asterisk MA-OWA operators are boundaries. From the forgoing equation, it became clear that for any F_{MA}

$$\min(a_i) \leqslant F_{\mathrm{MA}}(a_1, a_2, \ldots, a_n) \leqslant \max(a_i)$$

PROPERTY 2. *Commutative*: *The MA-OWA operator can be seen to be commutative. Let* $\langle a_1, a_2, \ldots, a_n \rangle$ *be a bag of aggregates, and let* $\langle d_1, d_2, \ldots, d_n \rangle$ *be a permutation of the* a_i. *Then,*

$$F_{\mathrm{MA}}(a_1, a_2, \ldots, a_n) = F_{\mathrm{MA}}(d_1, d_2, \ldots, d_n)$$

PROPERTY 3. *Monotonicity*: *A third characteristic associated with the operators is monotonicity. Assume* a_i *and* c_i *are a collection of aggregates,* $i = 1, \ldots, n$ *such that for each* i, $c_i \leqslant a_i$. *Then,*

$$F_{\mathrm{MA}}(c_1, c_2, \ldots, c_n) \leqslant F_{\mathrm{MA}}(a_1, a_2, \ldots, a_n)$$

where the frequency vector is the same to both vectors.

PROPERTY 4. *Idempotency*: *Another characteristic associated with these operators is idempotency. If* $a_i = a$ *for all* $i = 1, \ldots, n$, *then for any OWA operator*

$$F_{\mathrm{MA}}(a, a, \ldots, a) = a$$

PROPERTY 5. *Simplification*: *A new characteristic associated with the operators is simplification. Let* a_i *be a collection of aggregates* $i = 1, \ldots, n$, *and* $\delta_i > 1$ *for all* $i = 1, \ldots, n$. *Then,*

$$F_{\mathrm{MA}}(a_1, a_2, \ldots, a_n) = F_{\mathrm{MA}}(c_1, c_2, \ldots, c_n)$$

where the new δ_i *associated with* c_i *is calculated as* $\delta_i = \boldsymbol{\delta}_i - \min\{\delta_i\} - 1$ for all i $= 1, \ldots,$ n.

4.2. Example

Let $\langle 0.3, 0.4, 0.4, 0.4, 0.4, 0.6, 0.6 \rangle$ be a bag of aggregates. In this example, the elements to aggregate are ordered. In this case, the functions h_k is calculated as

$$h_k(b_1, b_2, \ldots, b_n) = \begin{cases} \sum_{j=1}^{n} p_{kj} & \text{if } k = 1 \\ \sum_{j=1}^{n-k+1} p_{kj} + 1 & \text{otherwise} \end{cases}$$

$$p_{kj} = \begin{cases} 1 & \text{if } b_j = b_{j+k-1} \text{ and } b_j \neq b_{j-1} \\ 1 & \text{if } j = 1 \text{ and } k \geqslant 1 \text{ and } b_j = b_k \\ 0 & \text{otherwise} \end{cases}$$

then

$$F_{\mathrm{MA}}(0.3, 0.4, 0.4, 0.4, 0.4, 0.6, 0.6) = 0.3 \cdot 0.027 + 0.4 \cdot 0.027 + 0.4 \cdot 0.083 + 0.4 \cdot 0.25 + 0.4 \cdot 0.5 + 0.6 \cdot 0.027 + 0.6 \cdot 0.083 = 0.417$$

where

$$h_1(0.3, 0.4, 0.4, 0.4, 0.4, 0.6, 0.6) = p_{11} + p_{12} + p_{13} + p_{14} + p_{15} + p_{16} + p_{17} = 3$$

$$p_{11} = 1; \quad p_{12} = 1; \quad p_{13} = 0; \quad p_{14} = 0; \quad p_{15} = 0; \quad p_{16} = 1; \quad p_{17} = 0$$

$$h_2(0.3, 0.4, 0.4, 0.4, 0.4, 0.6, 0.6) = p_{21} + p_{22} + p_{23} + p_{24} + p_{25} + p_{26} + 1 = 3$$

$$p_{21} = 0; \quad p_{22} = 1; \quad p_{23} = 0; \quad p_{24} = 0; \quad p_{25} = 0; \quad p_{26} = 1$$

$$h_3(0.3, 0.4, 0.4, 0.4, 0.4, 0.6, 0.6) = p_{31} + p_{32} + p_{33} + p_{34} + p_{35} + 1 = 2$$

$$p_{31} = 0; \quad p_{32} = 1; \quad p_{33} = 0; \quad p_{34} = 0; \quad p_{35} = 0$$

$$h_4(0.3, 0.4, 0.4, 0.4, 0.4, 0.6, 0.6) = p_{41} + p_{42} + p_{43} + p_{44} + 1 = 2$$

$$p_{41} = 0; \quad p_{42} = 1; \quad p_{43} = 0; \quad p_{44} = 0$$

$$h_5(0.3, 0.4, 0.4, 0.4, 0.4, 0.6, 0.6) = p_{51} + p_{52} + p_{53} + 1 = 1$$

$$p_{51} = 0; \quad p_{52} = 0; \quad p_{53} = 0$$

$$h_6(0.3, 0.4, 0.4, 0.4, 0.4, 0.6, 0.6) = p_{61} + p_{62} + 1 = 1$$

$$p_{61} = 0; \quad p_{62} = 0$$

$$h_7(0.3, 0.4, 0.4, 0.4, 0.4, 0.6, 0.6) = p_{71} + 1 = 1$$

$$p_{71} = 0$$

The h_i's values are $\langle 3, 3, 2, 2, 1, 1, 1 \rangle$, and the g_j's are $\langle 1, 1, 2, 3, 4, 1, 2 \rangle$. Then, the w_i's are

$$w_1 = \frac{1}{3 \cdot 3 \cdot 2 \cdot 2 \cdot 1 \cdot 1 \cdot 1} = 0.027, \quad w_2 = \frac{1}{3 \cdot 3 \cdot 2 \cdot 2 \cdot 1 \cdot 1 \cdot 1} = 0.027$$

$$w_3 = \frac{1}{3 \cdot 2 \cdot 2 \cdot 1 \cdot 1 \cdot 1} = 0.083, \quad w_4 = \frac{1}{2 \cdot 2 \cdot 1 \cdot 1 \cdot 1} = 0.25$$

$$w_5 = \frac{1}{2} = 0.5, \quad w_6 = \frac{1}{3 \cdot 3 \cdot 2 \cdot 2 \cdot 1 \cdot 1 \cdot 1} = 0.027$$

$$w_7 = \frac{1}{3 \cdot 2 \cdot 1 \cdot 1 \cdot 1} = 0.083$$

5. CONCLUSIONS

In this study, we propose a new neat OWA operator (MA-OWA) that is based on the majority process. This operator solves the distribution problems when the items to aggregate have cardinality greater than one. It is very useful as an aggregation operator from a computation point of view, because they may be implemented as tables or simple procedures.

In addition, we define the procedure for calculating the weights as a function of the cardinality of the elements. Also, we present some properties of the operator.

References

1. Yager R. On ordered weighted averaging aggregation operators in multi-criteria decision making. IEEE Trans Syst Man Cybern 1988;18:183–190.
2. Filev D, Yager R. On the issue of obtaining OWA operator weights. Fuzzy Sets and Systems 1996;94:157–169.
3. Fodor J, Marichal J-L, Roubens M. Characterization of the ordered weighted averaging operators. IEEE Transac Fuzzy Syst 1995;3(2):236–240.
4. Schaefer PA, Mitchell HB. A generalized OWA operator. Intl J Intell Syst 1999;14:123–143.
5. Yager R. New modes of OWA information fusion. Int J Intell Syst 1998;13:661–681.
6. Yager R, Kacprzyk J. The ordered weighted averaging operators. In: Theory and applications. Boston, MA: Kluwer Academic Publishers; 1997.
7. Yager R. OWA neurons: A new class of fuzzy neurons. In: Proc Int Joint on Neural Networks, Vol I. Baltimore; 1992. pp 226–231.
8. Yager R. Fuzzy quotient operators for fuzzy relational data bases. In: Proc Int Fuzzy Engineering Symposium. Yokohama, Japan; 1991. pp 289–296.
9. Yager R. Including importances in OWA aggregations using fuzzy systems modeling. IEEE Trans Fuzzy Syst 1998;6(2):286–294.
10. Yager R. Quantifier guided aggregation using OWA operators. Int J Intell Syst 1996;11: 49–73.
11. Yager R. Nonmonotonic OWA operators. Soft Comput 1999;3:187–196.
12. O'Hagan M. Aggregating template rule antecedents in real-time expert systems with fuzzy set logic. In: Proc 22nd Annual IEEE Asilomar Conf On Signals, Systems, and Computers. Pacific Grove, CA; 1988. pp 681–689.
13. Yager R, Filew DP. Fuzzy logic controllers with flexible structures. In: Proc 2nd Int Conf on Fuzzy Sets and Neural Networks. Izuka, Japan; 1992. pp 317–320.
14. Marimin M, Umano M, Hatono I, Tamura H. Linguistic labels for expressing fuzzy preference relations in fuzzy group decision making. IEEE Trans Syst Man Cybern 1998;28(2):205–218.
15. Mitchell HB, Schaefer PA. Multiple priorities in an induced ordered weighted averaging operator. Int J Intell Syst 2000;15:317–327.
16. Vicenc T. The weighted OWA operator. Intl J Intell Sys 1997;12:153–166.
17. Herrera F, Herrera-Viedma E, Verdegay JI. Direct approach processes in group decision making using linguistic OWA operators. Fuzzy Sets Syst 1996;79:175–190.
18. Yager R. Families of OWA operators. Fuzzy Sets Syst 1993;59:125–148.
19. Robertson G, Webb G. Cake-cutting algorithms: Be fair if you can. Portland, OR: A. K. Peters Ltd.; 1998.
20. Yager R. Fuzzy quotient operator. In: Proc 4th Int Conf On Information Processing a Management of Uncertainty. Palma de Majorca, Spain; 1992. pp 317–322.