and executes iterations until the maximal implication chain length has been spanned. It prints out all initial and final fuzzy truth values.

REFERENCES

[1] T. Agerwala, "Putting Petri nets to work," *IEEE Computer*, pp. 85–94, Dec. 1979.

[2] H. Atabakhche, D. Barbalho, R. Valette, and M. Courvoisier, "From Petri net based PLC's to knowledge based control," in *Proc. IECON'86*, Jan. 1986, pp. 817–822.

[3] D. J. Burr, "A neural network digital recognizer," in *Proc. 1986 Conf. Syst. Man Cybern.*, Atlanta, GA, Aug. 1986, vol. 2, pp. 1621–1625.

[4] G. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Comput. Vision, Graphics, Image Processing*, vol. 37, pp. 54–115, Jan. 1987.

[5] B. R. Gaines and M. L. Shaw, "From fuzzy logic to expert systems, *Inform. Sci.*, vol. 36, pp. 5–15, 1985.

[6] A. W. Holt and F. Commoner, *Events and Condition*. New York: Applied Data Research, 1970; also in *Rec. Project MAC Conf. Concurrent Syst. and Parallel Computation*. New York: ACM, 1970, pp. 1–52.

[7] J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA*, vol. 79, pp. 2554–2558, 1982.

[8] C. G. Looney and A. R. Alfize, "Logical controls via Boolean rule matrix transformations," *IEEE Trans. Syst. Man Cybern.*, vol. 17, no. 6, Nov./Dec. 1987.

[9] C. G. Looney, "Expert control design with fuzzy rule matrices, *Int. J. Expert Syst.: Res. Appl.*, in press.

[10] J. L. Peterson, *Petri Nets, Theory and the Modelling of Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1981.

[11] C. A. Petri, "Kommunikation mit Automaten," Schriften des Rheinish-Westfalischen Institutes für Instrumentelle Mathematik an der Universtät Bonn, Heft 2, Bonn, W. Germany, 1962.

[12] M. Sugeno, "An introductory survey of fuzzy control," *Inform. Sci.*, vol. 36, pp. 59–83, 1985.

[13] D. Tabak, "Petri net representation of decision models," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-15, no. 6, pp. 812–818, Nov./Dec. 1985.

[14] M. G. Thomason, "Convergence of powers of a fuzzy matrix," *J. Math. Anal. Appl.*, vol. 57, pp. 476–480, 1977.

[15] J. Warfield, "Binary matrices in system modelling," *IEEE Trans. Syst. Man Cybern.*, vol. SCM-3, no. 5, pp. 441–449, Sept. 1973.

[16] M. D. Zisman, "Use of production systems for modelling asynchronous, concurrent processes," in *Pattern Directed Inference Systems*, D. A. Waterman and F. Hayes-Roth, Eds. New York: Academic, 1978, pp. 53–68.

On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking

RONALD R. YAGER

Abstract **—We are primarily concerned with the problem of aggregating multicriteria to form an overall decision function. We introduce a new type of operator for aggregation called an ordered weighted aggregation (OWA) operator. We investigate the properties of this operator. We particularly see that it has the property of lying between the "and," requiring all the criteria to be satisfied, and the "or," requiring at least one of the criteria to be satisfied. We see these new OWA operators as some new family of mean operators.**

INTRODUCTION

The problem of aggregating criteria functions to form overall decision functions is of considerable importance in many disciplines. A primary factor in the determination of the structure of such aggregation functions is the relationship between the criteria involved. At one extreme is the situation in which we desire that all the criteria be satisfied. At the other extreme is the case in which the satisfaction of any of the criteria is all we desire. These two extreme cases lead to the use of "and" and "or" operators to combine the criteria functions.

Manuscript received June 1, 1987; revised September 28, 1987.

The author is with the Machine Intelligence Institute, Iona College, New Rochelle, NY 10801.

IEEE Log Number 8718791.

Our purpose in this paper is to introduce a new family of operators called ordered weighted averaging (OWA) operators that provide an aggregation which lies in between these two extremes. The name "orand" operator may be more appropriate. We shall see the simple mean is a special case of this new operator. We should carefully point out this operator is different than the classical weighted average in that coefficients are not associated directly with a particular attribute but rather to an ordered position. We shall further see that the structure of these operators are very much in the spirit of combining the criteria under the guidance of a quantifier. That is, the requirement that "most" of the criteria be satisfied corresponds to one of these OWA operators. We can see this work very much in the spirit of [1], [2] where Yager discusses an alternative approach to these aggregation processes.

FORMULATING OF THE AGGREGATION PROBLEM

Assume $A_1, A_2, \cdots, A_n$ are n criteria of concern in a multicriteria problem. Let X be some proposed alternative. For each criteria, A_j, $A_j(x) \in [0,1]$ indicates the degree to which x satisfies that criteria. We shall use I to indicate the unit interval, thus $A_j(x) \in I$. Our central interest is the problem of formulating an overall decision function D such that for any alternative x, $D(x) \in I$ indicates the degree to which x meets our desired requirements with respect to the criteria.

The problem becomes that of formulating a function D from the constituent individual criteria functions

$$D(x) = F(A_1(x), A_2(x), \cdots, A_n(x)).$$

The structure of F should be such that the following conditions are met.

1) As our satisfaction in the individual alternative increases the overall satisfaction should increase; if $A_j(x) \geqslant A_j(y)$ for all j then $D(x) \geqslant D(y)$. We call this a monotonicity property or positive association of individual criteria with aggregate preferences.

2) The equality of importance of the different criteria means that F should be symmetric with respect to the criteria. More specifically, if $a_1, \cdots, a_n$ is a collection of numbers in the unit interval than any one to one association of these numbers with the $A_j(x)$'s will result in the same value for $D(x)$, that is if $n = 3$

$$F(a_1, a_2, a_3) = F(a_2, a_1, a_3).$$

More formally recalling that a bag [3] is a set like object which allows duplication but pays no attention to ordering then

$$D(x) = F(\langle A_1(x), A_2(x), \cdots, A_n(x) \rangle)$$

where we use $\langle$ and $\rangle$ to denote a bag. We shall call this property symmetry or generalized commutativity.

Another consideration that we must be concerned with in formulating is the interrelationship between the criteria which we desire to model.

At one extreme is the situation in which we desire that an alternative satisfy "all" the criteria. In this case we see that x must satisfy A_1 and A_2 and $A_3, \cdots$, and A_n. Thus the requirement that all the conditions be satisfied is manifested by an "anding" of the criteria values.

At the other extreme is the situation in which we desire that an alternative satisfy "at least one of the criteria." In this case we desired that x satisfy A_1 or A_2 or $A_3, \cdots$, or A_n. Thus the requirement that *at least one* of the criteria be satisfied is manifested by an "oring" of the criteria values in the formulation of the decision function.

In many cases the interrelationship between the criteria lies somewhere between these two extreme cases of wanting "all" or "at least one." That is, we desire that "most" or "many" or "at least half" or "more than four" of the criteria are satisfied. It is our purpose here to obtain a general functional form of this type of situation.

General "Anding" and "Oring" Operators

There exists a class of operators called t-norms [4]–[6] that provide way of quantitatively implementing the type of "anding" aggregation implied by the "all" requirement. A closely related class of operators, called co-t-norms, provide a way of implementing the type "oring" operating previously discussed. In this section we briefly discuss these operators and point out some properties relevant to our discussion.

A t-norm T is a mapping

$$T: [0,1]\times[0,1] \to [0,1]$$

such that

1) $T(a,b) = T(b,a)$ "commutative",
2) $T(a,b) \geqslant T(c,d)$ if $a \geqslant c$ and $b \geqslant d$ monotonic,
3) $T(a,T(b,c)) = T(T(a,b),c)$ "associative",
4) $T(1,a) = a$.

Among these operators that satisfy the property of being a t-norm are

1) $T(a,b) = \text{Min}(a,b)$,
2) $T(a,b) = a\cdot b$,
3) $T(a,b) = 1-\text{Min}(1,((1-a)^P+(1-b)^P)^{1/P})$ for $P\geqslant 1$ [7].

Bonnisone [8] among others has looked with considerable detail into the empirical properties of t-norm operators.

We should note that while the t-norms were defined in terms of binary operators they were extendible, via their associative property, to combining any number of values in the unit interval. Thus if the relationship between the criteria is an "anding" then

$$D(x) = T(A_1(x), A_2(x),\cdots, A_n(x_n))$$

where T is some t-norm operator. The issue of selection of the appropriate t-norm in a given situation is one discussed by Bonnisone [8] and Yager [9].

An important property of the t-norm operator is stated in the following theorem.

Theorem: Assume T is any t-norm operator; then for any a and b

$$T(a,b) \leqslant \text{Min}(a,b).$$

Proof: Without loss of generality assume $\text{Min}(a,b) = b$. Since

$$T(1,b) = b$$

and for any a, $a \leqslant 1$; then

$$T(a,b) \leqslant T(1,b) \leqslant b \leqslant \text{Min}(a,b).$$

An implication of this theorem is that the t-norm Min provides the largest of these class of operators. We note that it is easy to show that for any collection $a_1, a_2,\cdots, a_n$

$$T(a_1,\cdots,a_n) \leqslant \text{Min}(a_1,\cdots,a_n).$$

This result implies that in multicriteria decisionmaking the use of an "anding" allows for no compensation for one bad satisfaction. Another interesting and unique property of the Min operator is that it is the only T norm-operator such that for all $a\in I$

$$T(a,a) = a.$$

We say it has the idempotency property. We should point out that the conditions one, two, and three of the defining definition of t-norms essentially provide the satisfaction to requirements I & II, symmetry (generalized commutativity) and monotonicity required of aggregation operators. We note that it is condition 4, $T(1,a) = a$, that essentially stipulates this as an "anding" operator by requiring a form of "allness" satisfied by a t-norm.

A co-t-norm S is a mapping

$$S: [0,1]\times[0,1] \to [0,1]$$

such that

1) $S(a,b) = S(b,a)$ "commutative"
2) $S(a,b) \geqslant S(c,d)$ if $a\geqslant c$ and $b \geqslant d$ monotonic,
3) $T(a,T(b,c)) = T(T(a,b),c)$ "associative",
4) $T(0,a) = a$, "at least oneness."

Among those operators that satisfy the property of being a co-t-norm are

1) $S(a,b) = \text{Max}(a,b)$,
2) $S(a,b) = a+b-a\cdot b$,
3) $S(a,b) = \text{Min}[1, a^P+b^P)^{1/P}]$ for $p\geqslant 1$ [7].

Thus if the relationship between the criteria is a pure "oring," then

$$D(x) = S(A_1(x), A_2(x),\cdots, A_n(x))$$

where S is some co-t-norm. An important property for these co-t-norms is

Theorem: Assume S is any co-t-norm operator; then for any a and b,

$$S(a,b) \geqslant \text{Max}(a,b).$$

An implication of this theorem is that the Max provides the smallest of these class of operators. It can be shown that for any collection $a_1,\cdots,a_n$

$$S(a_1,\cdots,a_n) \geqslant \text{Max}(a_1,\cdots,a_n).$$

This implies that in multicriteria decisionmaking the use of a pure "oring" allows for no distraction from one good satisfaction. We should also point out that Max is the only co-t-norm having the idempotency property, for all $a\in I$,

$$S(a,a) = a.$$

Again it should be noted that it is condition 4 that makes this an "or" operation by implementing an "at least one" type of condition. Conditions 1, 2, and 3 again just enforce the symmetry and monotonicity conditions.

OWA Operators

In many cases of formulation of multiple criteria decision functions the type of aggregation implicitly desired by a decision maker is neither the pure "anding" of the t-norm with its complete lack of compensation nor the pure "oring" of the S operator with its complete submission to any good satisfaction as well as its indifference to the individual criteria. In many cases the type of aggregation operator desired lies somewhere between these two extremes. In this section, we shall introduce a new type of operator that we shall call an ordered weight averaging (OWA) operator. We shall see that this new aggregation operator allows us to easily adjust the degree of "anding" and "oring" implicit in the aggregation. As we shall see, a more descriptive name for this operator may be an "orand" operator because of its acting like a combination of the two.

Definition: A mapping F from

$$I^n \to I \ (\text{where } I = [0,1])$$

is called an OWA operator of dimension n if associated with F, is a weighting vector $\boldsymbol{W}$,

$$\boldsymbol{W} = \begin{bmatrix} W_1 \\ W_2 \\ W_n \end{bmatrix}$$

such that

1) $W_i \in (0,1)$
2) $\Sigma_i W_i = 1$

and where

$$F(a_1, a_2, \cdots, a_n) = W_1 b_1 + W_2 b_2 + \cdots W_n b_n,$$

where b_i is the ith largest element in the collection $a_1, a_2, \cdots, a_n$.

We shall call an n vector $\boldsymbol{B}$ an ordered argument vector if each element $b_i \in [0,1]$ and $b_i \geqslant b_j$ if $j > i$. Given an OWA operator F with weight vector $\boldsymbol{W}$ and an argument tuple $(a_1, a_2, \cdots, a_n)$ we can associate with this tuple an ordered input vector $\boldsymbol{B}$ such that $\boldsymbol{B}$ is the vector consisting of the arguments of F put in descending order. Using this notation then

$$F(a_1, \cdots, a_n) = W'B,$$

the inner product of W' and B. We shall sometimes find it convenient to denote $F(a_1, \cdots, a_n)$ as $F(B)$ where B is the associate ordered argument vector.

It is important to emphasize the fact that the weights, the W_i's, are associated with a particular ordered position rather than a particular element. That is W_i is the weight associated with the ith largest element whichever component it is.

We note that it can easily be shown that for any ordered argument vector B and any OWA operator F with weighting vector $\boldsymbol{W}$ that

$$0 \leqslant F(B) \leqslant 1.$$

The following simple example illustrates the use of these OWA operators.

Example: Assume F is an ordered weighting averaging operator of size $n = 4$ with weighting vector,

$$\boldsymbol{W} = \begin{bmatrix} 0.2 \\ 0.3 \\ 0.1 \\ 0.4 \end{bmatrix}$$

a) Calculate $F(0.6, 1, 0.3, 0.5)$.
In this case the ordered argument vector $\boldsymbol{B}$ is

$$\boldsymbol{B} = \begin{bmatrix} 1.0 \\ 0.6 \\ 0.5 \\ 0.3 \end{bmatrix},$$

hence

$$F(0.6, 0.1, 0.3, 0.5)$$

$$= F(B) = W'B = [0.2, 0.3, 0.1, 0.4] \begin{bmatrix} 1.0 \\ 0.6 \\ 0.5 \\ 0.3 \end{bmatrix}$$

$$= (0.2)(1) + (0.3)(0.6) + (0.1)(5) + (0.3)(0.4) = 0.55$$

b) Calculate $F(0, 0.7, 0.1, 0.2)$.
Here

$$B = \begin{bmatrix} 1.0 \\ 0.7 \\ 0.2 \\ 0.0 \end{bmatrix}$$

and therefore

$$F(B) = [0.2, 0.3, 0.1, 0.4] \begin{bmatrix} 1.0 \\ 0.7 \\ 0.2 \\ 0.0 \end{bmatrix} = 0.43$$

Properties of OWA Operators

In this section we shall investigate some of the properties of these new operators. We first recall that in our discussion of aggregation operators we required two fundamental properties that any aggregation operator must satisfy.

Our first theorem shows that these OWA operators are monotonic with respect to argument values.

Theorem: Assume F is an OWA operator. Let $A = [a_1, \cdots, a_n]$ be an ordered argument vector. Let $B = [b_1, \cdots, b_n]$ be a second ordered argument vector such that for each j

$$a_j \geqslant b_j$$

then $F(A) \geqslant F(B)$.

Proof: Since

$$F(A) = W'A$$

and

$$F(B) = W'B$$

the result follows directly from the property $a_j \geqslant b_j$.

Corollary: $F(c_1, c_2, \cdots, c_n) \geqslant F(d_1, d_2, \cdots, d_n)$ if $c_j \geqslant d_j$.

Proof: Let A and B be the ordered argument vectors in each of these cases. Then it can easily be shown that $a_j \geqslant b_j$.

We also note that every operator of this type exhibits the kind of symmetry implied by equal importance of criteria.

Theorem: Assume F is an OWA operator. Then

$$F(a_1, a_2, \cdots, a_n) = F(a_1', a_2', \cdots, a_n')$$

where

$$(a_1', a_2', \cdots, a_n')$$

is any permutation of the elements in $(a_1, \cdots, a_n)$.

Proof: If B and B' are the ordered argument vectors of $(a_1, \cdots, a_n)$ and (a_1', a_2', a_n') respectively then $B = B'$. Hence $F(B) = F(B')$.

Thus we see that these operators exhibit the kind of generalized commutativity we desire.

We note one further and perhaps defining property of these OWA operators.

Theorem: All OWA operators are idempotent in the sense that if $a_j = a$, for all $j = 1, \cdots, n$, then

$$F(a_1, a_2, \cdots a_n) = a.$$

It should be noted that the desire to satisfy the condition of symmetry (generalized commutativity) is what forced us to use an "ordered" weighted average rather than simply taking a weighted average. For example, if we defined

$$G(a_1, \cdots, a_n) = \sum_{j=1}^{n} w_j a_j$$

we would not in general get the symmetry condition satisfied.

Example: Assume

$$G(a_1, a_2) = 0.7a_1 + 0.3a_2.$$

If $a_1 = 1$ and $a_2 = 0$ then

$$G(1,0) = 0.7.$$

However, if $a_1 = 0$ and $a_2 = 1$ then

$$G(0,1) = 0.3$$

thus $G(1,0) \neq G(0,1)$.

One special OWA operator worth noting is the pure "averaging" or "mean" operator. In this case $W_j = 1/n$ and thus

$$F(B) = 1/n \sum_{j=1}^{n} b_j.$$

We shall denote this as F_M. It should be noted that this is the only case of a "fixed" weighting operator that is also an OWA operator.

We shall now introduce two special weighting vectors associated with OWA operators.

Definition: W_* is defined as the weighting vector that has $W_n = 1$ and $W_j = 0$ for all $j \neq n$. W^* is defined as the weighting vector which has $W_1 = 1$ and $W_j = 0$ for all $j \neq 1$.

Theorem: Assume B is an arbitrary ordered input vector. Then for any weighting vector W

$$(W_*)'B \leqslant W'B \leqslant (W^*)'B.$$

Proof: a) $(W_*)'B \leqslant W'B$

$$(W_*)'B = b_n$$

$$W'B = \sum_n W_j b_j = b_n W_n + \sum_{j=1}^{n-1} W_j b_j.$$

Since B is an ordered input vector then $b_j \geqslant b_k$ for $k > j$. In particular $b_n < b_j$ for $j = 1, \cdots, n-1$, hence

$$W'B \geqslant b_n W_n + b_n * \sum_{j=2} W_j,$$

however

$$\sum_{j=1}^{n-1} W_j = 1 - W_n,$$

thus

$$W'B \geqslant b_n W_n + (1 - W_n) b_n \geqslant b_n \geqslant (W_*)'B$$

b) $(W^*)'B \geqslant W'B$

$$(W^*)'B = b_1$$

$$W'B = \sum_n W_j b_j = \sum_{j=2}^{n} W_j b_j + b_1 W_1.$$

Since $b_1 \geqslant b_j$ then

$$W'B \leqslant b_1 W_1 + b_1 \sum_{j=2}^{n} W_j \leqslant b_1.$$

We shall use F^* and F_* to denote the OWA operator with respectively W^* and W_* as their weighting vectors. If $A = \langle a_1, a_2, \cdots, a_n \rangle$ is a bag of criteria values, then for any operator F,

$$F_*(A) \leqslant F(A) \leqslant F^*(A).$$

Thus F_* and F^* provide a lower and upper bound on the aggregation using an ordered weighted average operator.

Theorem: Assume $a_1, \cdots, a_n$ is a collection of numbers each lying in the unit interval then

$$F_*(a_1, \cdots, a_n) = \mathrm{Min}_j(a_j)$$

$$F^*(a_1, \cdots, a_n) = \mathrm{Max}_j(a_j.)$$

Proof: a) $F_*(a_1, \cdots, a_n) = (W_*)'B = b_n$
but $b_n = \mathrm{Min}_j(a_j)$.
b) Follows in a similar manner.

Thus the two extreme cases of OWA operators are the "and" and "or" operators. In particular, the largest F operator is the smallest "or" operator, Max, while the smallest F operator is the largest "and" operator, Min.

Assume $A = [a_1, \cdots, a_n]$ is a collection of attribute satisfactions. Let T and S be any t and co-t-norms. Let F be any OWA operator, then

$$T(A) \leqslant F(A) \leqslant S(A).$$

Thus F provides an aggregation type operator that always lies between the "and" and the "or" aggregation. This property leads us to think of these OWA operators as a kind of "orand" operator.

A natural question that arises in the formulation of an aggregation function F of the type we have just proposed concerns itself with the issue of obtaining the weights associated with the weighting vector, the W_i's. For our purposes we shall consider that F is a function used to aggregate n criteria.

There exists at least two ways that can be used to obtain the value of the W_i's. The first approach is to use some kind of learning mechanism. In this approach we use some sample data, arguments and associated aggregated values and try to fit the weights to this collection of sample data. The process involves the use of some kind of regression model.

A second approach is to try to give some semantics or meaning to the W_i's. Then based upon these semantics we can have the decisionmaker directly provide the values for the W_i's. This approach also will provide some further insight into the meaning of the OWA operators we have just introduced. In the following we shall provide some semantics for the weights.

We shall let

$$S_K = \sum_{j=0}^{K} W_j.$$

We note that $S_n = 1$ and $S_o = 0$. Assume we have an input vector of criteria satisfaction B such that $b_j = 1$ for $j \leqslant K$ and $b_j = 0$ for $j > K$. This indicates that K of the criteria are completely satisfied and the rest are completely unsatisfied. In this situation

$$F(B) = W'B = \sum_{j=1}^{K} W_j = S_K.$$

Thus S_K is the degree of satisfaction the decisionmaker has if he satisfies K/N portion of the criteria. Furthermore since $S_K = S_{K-1} + W_K$ we can interpret W_K as the degree of additional (or marginal) satisfaction he gets when we go from satisfaction of $K-1$ of the criteria to the satisfaction of K. We note that in this interpretation then the case of $W_j = 1/n$ corresponds to a linear increase for each increment. From a pragmatic point of view it appears more natural for the decisionmaker to provide the S_k function, the degree to which he is happy with k criteria being satisfied. We note that it is easy to obtain the W_*'s from the S_K's since

$$W_K = S_K - S_{K-1},$$

where $S_o = 0$.

With this interpretation F_* indicates the situation where there is no satisfaction until all the criteria are satisfied while F^* implies complete satisfaction if at least one of the criteria are satisfied.

Quantifiers and OWA Operators

Drawing upon Zadeh's [10], [11] concept of linguistic quantifiers and Yager's application [1], [2], [12] of this idea to multicriteria decision making we can provide a deeper and more unifying interpretation of the weighting function W associated with an aggregation operator F.

The classic binary logic allows for the representation of two quantifiers, "there exists" and "for all." In natural language we use many additional quantifiers such as "almost all," "few," "many," "most," etc. the theory of approximate reasoning [13] extends the binary logic among other ways by allowing for the representation of these linguistic quantifiers. In [10] Zadeh suggests that quantifiers are of at least two kinds—those which say something about the number of elements and those which say something about the proportion of elements. It is also suggested by Zadeh that quantifiers can be represented as fuzzy subsets of either the unit interval or the real line. The distinction is based upon whether the quantifier relates to an absolute or is a proportion type statement. Thus if Q is relative a quantity such as "most," then Q can be represented as a fuzzy subset of I such that for each $r \in I$, $Q(r)$ indicates the degree to which r portion of the objects satisfies the concept denoted by Q. We note that the quantifier "for all" can be represented as a fuzzy subset of I such that

$$Q(1) = 1$$
$$Q(r) = 0 \qquad r \neq 1.$$

We shall call a quantifier Q monotonically nondecreasing if $r_1 > r_2 \rightarrow Q(r_1) \geqslant Q(r_2)$.

Assume we have a decision problem in which we have n criteria, $A_1, \cdots, A_n$, where $A_j(x)$ indicates the degree to which alternative x satisfies criteria A_j. Furthermore, assume in specifying the manner in which these criteria are to be aggregated to form an overall decision function the decisionmaker states that he desires Q of the criteria be satisfied. In this representation Q is an absolute quantifier definable on the space $L \equiv [O, n]$. In addition for $y \in L$, $Q(y)$ indicates the degree to which the decisionmaker is satisfied with y criteria being solved. We can make the following observations:

1) $Q(O) = 0$, the decisionmaker gets absolutely no satisfaction if he gets no criteria satisfied.

2) $Q(n) = 1$, he is completely satisfied if he gets all the criteria satisfied.

3) If $r_1 > r_2$, then $Q(r_1) \geqslant Q(r_2)$ as he gets more criteria satisfied he will not become less satisfied.

One can see that this quantifier function has the same properties as the function of the previous section. Therefore it is our conjecture that the weighing vector W is a manifestation of the quantifier underlying the aggregation process. In particular, if a decisionmaker suggests that they want Q of the objectives satisfied, then we obtain the weighing vector as $W_K = Q(K) - Q(K-1)$, $K = 1, \cdots, n$, and $Q(0) = 0$.

On the other hand if the weights are obtained via some kind of learning process, we can conjecture an underlying quantifier Q as

$$Q(K) = \sum_{j=1}^{K} W_1.$$

The important idea of this section is that the type of OWA operators we have been discussing appear to be manifestations of monotonic quantifiers. We also recall that "and" (for all) operator that corresponds to one of the extremes of these quantifiers has

$$W_n = 1$$
$$W_i = 0 \qquad i \neq n.$$

In particular

$$Q_{\text{"and"}}(K) = 0 \qquad k \neq n$$
$$Q_{\text{"and"}}(K) = 1.$$

On the other hand, for the "or" (there exists) quantifier

$$W_j = 1 \qquad \text{for } j = 1$$
$$W_j = 0 \qquad \text{for } j \neq 1,$$

therefore,

$$Q_{\text{"or"}}(K) = 1 \qquad K \geqslant 1.$$

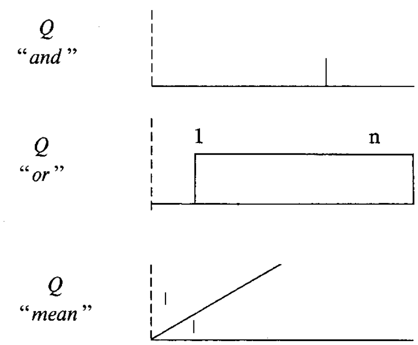
The pure averaging quantifier has

$$W_j = 1/n \qquad \text{for all } j = 1, \cdots, n,$$

then

$$Q_{\text{"mean"}}(K) = K/n,$$

its a linear quantifier.



Thus we see that the weights associated with the OWA function determine the kind of quantifier it is effecting. By varying the assignment of the weights in W we can move from a Min type, "for all," quantifier, to a Max type there exists quantifiers. In particular, we can capture aggregations which emulate things like "most" etc. Thus we see that these OWA operators provide an interesting class of operators.

Measure of "Andness" and "Orness"

Assume F is an OWA operator with weighting function W. If $W_n = 1$ then as we have shown F is a pure "and" operator while if $W_1 = 1$, then F is a pure "or" operator. We can further observe the closer all the total weight is to being in W_1 the closer the F function is to being a pure "or" operator while the closer it is to being all in W_n the function is closer to an "and." We shall here introduce a measure of "orness" associated with a weighting function.

Definition: Assume F is an OWA aggregation operator with weighing function $W = [W_1, \cdots, W_n]$. The degree of "orness" associated with this operator is defined as

$$\mathit{orness}(W) = (1/n-1) \sum_{i=1}^{n} ((n-i) * W_i).$$

Example: a) $W = [1\ 0\ 0\ 0 \cdots]$

$$\mathit{orness}(W) = 1$$

b) $W = [0\ 0\ 0 \cdots 1]$

$$\mathit{orness}(W) = 0$$

c) $W = (1/n, 1/n, \cdots 1/n)$

$$\begin{aligned} \mathit{orness}(W) &= 1/n - 1 \sum_{i=1}^{n} (n-i) 1/n \\ &= \frac{1}{n} * \frac{1}{(n-1)} * \sum_{i=1}^{n} n - \sum_{i=1}^{n} i \\ &= (1/n)(1/(n-1))(n - n - (n)(n+1)/2) \\ &= 1/2. \end{aligned}$$

We can see that this measure of "orness" is defined by

$$\text{"orness"}(W) = \sum_{j=1}^{n} (h_n(j) * W_j)$$

where $h_n(j)$ is a linear type function. That is

$$h_n(j) = (n/n-1) - (j/n-1) = (n-j)/(n-1), \qquad j = 1, \cdots, n$$

In point of fact since $h_n(i) > h_n(j)$ for $j > i$ then we can see that h_n is really a prototype "linear argument vector." Thus the measure of "orness" of an OWA operator is its aggregated value under a linear argument vector. It is interesting to point out whatever value n is, $h_n(n) = 0$ and $h_n(1) = 1$. Thus given this type of input any S-norm always evaluates to degree of "orness" equal to 1 and every t-norm to zero "orness."

We recall that if $W = [W_1, \cdots W_n]$ then we define

$$Q_k = \sum_{j=1}^{k} W_j,$$

where $Q_n = 1$ and $Q_K \geqslant Q_{K-1}$.

Theorem: Assume W and W' are two weighting functions such that for each K

$$Q_K \geqslant Q'_K,$$

then

$$\text{"orness}(W)\text{"} \geqslant \text{"orness"}(W').$$

Proof: We shall let

$$U(K) = 1/n - 1 \sum_{j=1}^{K} (n-j) \cdot W_j$$

$$U'(K) = 1/n - 1 \sum_{j=1}^{K} (n-j) W'_j$$

thus $\text{orness}(W) = U(n)$ and $\text{orness}(W') = U(n)$. We shall first show that if for any K

$$U(K) \geqslant U'(K), \qquad \text{where } U(K) = U'(K) + a.$$

It is impossible for $U(K+1) > U'(K+1)$.

$$U(K+1) = U(K) + ((n-K-1)/(n-1)) * W_{K+1}$$

$$U(K+1) = U'(K) + (1/(n-1)) * (n-K-1) * W'_{K+1}.$$

For $U'(K+1) > U(K+1)$ then

$$W'_{K+1} - W_{K+1} = (n-1)/(n-K-1) * a.$$

However since it is always required that

$$Q_{K+1} \geqslant Q'_{K+1}$$

then to get this much additional $W'_{K+1} - W_{K+1}$, it must be the case that

$$Q_K - Q'_K \geqslant ((n-1)/(n-K-1)) * a.$$

However, if this condition holds true then

$$U(K) - U'(K)$$
$$\geqslant ((n-K)/(n-1)) * ((n-1)/(n-K-1)) * a > a,$$

thus this is a contradiction.

Furthermore, since

$$U(1) = W_1$$

$$U'(1) = W'_1,$$

then since $Q(1) = W_1$ and $Q'(1) = \mathrm{W}'_1$ then

$$U(1) \geqslant U'(1).$$

This theorem allows us to directly compare quantifiers and tell whether one is more of an "or" than another.

We should note that if Q_1 and Q_2 are two linguistic quantifiers then we say $Q_1 < Q_2$ if $Q_1(x) \leqslant Q_2(x)$ for all x. Thus from the above theorem if Q_1 and Q_2 are two quantifiers underlying aggregation functions F_1 and F_2 then if $Q_1 < Q_2$ then F_2 is a more "orlike" aggregation. In particular, we see that the more specific the monotone quantifier underlying the aggregation process the more "and" like the aggregation.

We shall define a measure of "andness" associated with an OWA operator as the complement of the "orness" thus "andness"$(W) = 1 -$ "orness"(W).

Consider two weighing functions W_1 and W_2 where

$$W = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \text{ and } W_3 = \begin{bmatrix} 1/5 \\ 1/5 \\ 1/5 \\ 1/5 \\ 1/5 \end{bmatrix}.$$

We note that while both of these weighting functions have the same degree of "orness," 0.5, we can see that they are different in the sense that the first one is more volatile and uses less of the input. In order to capture this idea we introduce a measure of "dispersion" associated with a weighting function W.

Definition: Assume W is a weighting vector with elements $W_1 \cdots W_n$ then the measure of dispersion of W is defined as

$$\text{dispersion}(W) = -\sum_j W_j \ln W_j.$$

We note that since this dispersion is really a measure of entropy and thus the following properties are valid.

1) if $W_j = 1$ for some i then the dispersion is minimum and $\text{dispersion}(W) = 0$
2) the dispersion is maximum if $W_j = 1/n$ and in this case $\text{dispersion}(W) = \ln n$.

It is interesting to see that this measure of dispersion uses this Shannon information concept, for in a certain sense the more disperse the W the more of the information about the individual criteria is being used in the aggregation of the aggregate value.

Assume Q is the quantifier underlying an aggregation process with a weighting vector W. Since

$$Q(K) = \sum_{j=1}^{K} W_j$$

it appears that the concept of dispersion in the framework of weighting vectors is closely related to the concept of the fuzziness [4] in the underlying quantifier. In particular, a very crisp quantifier such as "for all," "there exists," "at least 50 percent" tend to have less disperse weighting vectors while fuzzier quantifiers such as many tend to have a more disperse weighting vector.

In a General Setting

It appears that these new OWA operators can be seen as a special family of a more general class of mean-like operators. We can call these generalized means or more descriptively "orand" operators. We shall denote these general operators as R operators and define them as having the following properties:

$$R: I^n \to I$$

such that:

1) R satisfies a generalized commutativity (symmetry)

$$R(a_1, a_2, \cdots, a_n) = R(b_1, b_2, \cdots, b_n)$$

if the bags $\langle a_1, \cdots, a_n \rangle$ and $\langle b_1, \cdots, b_n \rangle$ are equal;
2) R is monotonic;

$$R(a_1, a_2,, \cdots, a_n) \geqslant R(b_1, \cdots, b_n)$$

if $a_i \geqslant b_i$ for all $i = 1, 2, n$;
3) R is idempotent. For every $a \in I$,

$$R(a, a, a, \cdots, a) = a.$$

We note Dubois and Prade [15] looked at binary operators of this type fact that these operators lie between the "and" and "or" operators can be seen by the following theorem.

Theorem: For any R operator

$$\mathrm{Min}(a_1,\cdots,a_n) \leqslant R(a_1,\cdots,a_n) \leqslant \mathrm{Max}(a_1,\cdots,a_n)$$

Proof: a) Assume $b = \mathrm{Min}(a_1,\cdots,a_n)$. By property 3,

$$R(b,b,\cdots b) = b$$

however since for each i, $a_i \geqslant b$, then by property 1,

$$R(a_1,a_2,\cdots,a_n) \geqslant R(b,\cdots,b) \geqslant b$$

b) follows in a similar manner.

Another family of operators which are members of this general class of R (orand) operators are the generalized means introduced by Dyckhoff & Pedrycz [16], in this case,

$$R(a_1,\cdots,a_n) = \left(\sum_{j=1}^{n} 1/n(a_j)^P\right)^{1/p}$$

where $p \in [-\infty,\infty]$.

In the general case of these R operators and in the OWA family in particular the one nice property that is not required, hence not usually available, is the associativity property. The usefulness of this property is based upon its ability of helping us easily include additional data. However, as discussed by Dubois & Prade [15] associativity and idempotency don't usually coexist easily.

Building Consistent OWA Operators

Assume $A_1,\cdots,A_n$ are n criteria having satisfactions $a_1,\cdots,a_n$ under some x. Assume F is some OWA function with weighting vector W, where $W = [W_1,\cdots,W_n]$. Thus

$$F(x) = \sum_{j=1}^{n} W_j b_j$$

where b_1 equals the jth biggest element in the bag $\langle a_1,\cdots,a_n\rangle$.

Assume A_{n+1} is some additional criteria having satisfaction a_{n+1} under x. Assume we are interested in aggregating the $n+1$ criteria in a method that is consistent with the aggregation of the original n criteria. That is we must find some $n+1$ order aggregation function F^* and its associated weighting vector $W^* = [W_1^*,\cdots,W_n^*,W_{n+1}^*]$ that is in some sense consistent with F and its W.

We shall say W and W^* are q-consistent if three exists some monotonic quantifier on the unit interval such that W and W^* could have been drawn from this same quantifier.

We recall that Q is a monotonic type quantifier if:

1) $Q(1) = 1$;
2) $Q(0) = 0$;
3) $Q(r_1) \geqslant Q(r_2)$ for all $r_1, r_2 \in I$ where $r_1 > r_2$.

Theorem: Two weighting vectors W and W^* of size n and m respectively are q-consistent if

1) $\sum_{j=1}^{K} W_j \geqslant \sum_{j=1}^{k} W_j^* \qquad k = 1,\cdots,n;$

2) $\sum_{j=1}^{K} W_j \leqslant \sum_{j=1}^{k+1} W_j^* \qquad k = 1,\cdots,n-1;$ and

3) $\sum_{j=1}^{n} W_j = \sum_{j=1}^{n+1} W_j^* = 1.$

Proof: Assume W and W^* are two functions taken from some function Q. We note that from W we can specify $Q(K/n)$ for $k = 1,\cdots,n$, in particular,

$$Q(K/n) = \sum_{j=1}^{K} W_j \qquad k = 1,\cdots,n.$$

We also note that from W^* we can specify $Q(K/n+1)$ for $k = 1,\cdots,n+1$. In particular

$$Q(K/n+1) = \sum_{j=1}^{K} W_j^* \qquad k = 1,\cdots,n+1.$$

The following observations can be made for any $k = 1,\cdots,n,n+1$

$$k/n > k/n+1 \qquad k = 1,\cdots,n$$

and

$$k+1/n+1 > k/n \qquad k = 1,\cdots,n-1.$$

From the monotonicity property of Q we can see that

$$Q(k/n) \geqslant Q(k/n+1)$$
$$Q(k+1/n+1) \geqslant Q(k/n).$$

Example: Assume we have $W = [1,0]$ a pure "or" operator. Consider $W^* = [0,0,1]$ a "pure and." Are they q-consistent? The requirements are:

1) $W_1 \geqslant W_1^*$;
2) $W_1 + W_2 \geqslant W_1^* + W_2^*$;
3) $W_1 \leqslant W_1^* + W_2^*$;
4) $W_1 + W_2 = W_1^* + W_2^* + W_3^*$.

We note that condition 3 fails to hold since

$$1 > 0 + 0.$$

A necessary condition for W^* to be q-consistent with W is that $W_1^* + W_2^* = 1$. Thus $W^* = [a, 1-a, 0]$ is q-consistent.

Given an n dimension vector W one can set up as a linear programming problem that of finding the $n+1$ vector W^* that is q-consistent with W and is the most "orlike" or "andlike." Here our objective function would involve use of our measure of "orness."

Including Unequal Importances

Implicit in our methodology for aggregating criteria is the assumption that all criteria are of equal importance to the decisionmaker. In some cases the decision maker may assign differing degrees of importance to each of the criteria. In this section we shall suggest a scheme for including the ability to handle different importances in OWA operators. We note that in [17]–[21] we looked at issues related to importance.

Assume F is an OWA operator with weighting vector W. Assume A_1, $1 = 1,2,\cdots,n$, are a collection of criteria. Assume for each criteria $\alpha_1 \in I$ indicates the degree of importance associated with the criteria. For each criteria, let $A_1(x) \in I$ be the degree of satisfaction to criteria i. In this situation then our overall decision function evaluated at x is

$$F(x) = F(a_1,\cdots,a_n)$$

where

$a_j = H(A_j(x), \alpha_j)$ is the effective satisfaction to criteria A_j.

We further recall if b_K is the kth largest element in the bag $\langle a_1,\cdots,a_n\rangle$ then

$$F(x) = \sum_{K=1}^{n} b_k * W_K.$$

One possible form of the function H is

$$a_j = H(A_j(x), \alpha_j) = (\alpha_j \vee p) * (A_j(x))^{(\alpha j \vee q)}.$$

In the above formula q is the degree of "orness" associated with the weighting function W and p is its complement $p + q = 1$, the degree of "andness" associated with W. We first note if $p = 1$ and $q = 0$, F is a pure "and" operator, then $a_j = A_j(x)^{\alpha j}$.

Thus in this pure "and" environment, since $W_n = 1$, then

$$D(x) = \underset{j=1 \cdots n}{\text{Min}} A_j(x)^{\alpha j}.$$

This is in complete agreement with the original method for including importances suggested by Yager [17]. On the other hand if we have a pure "or" then $p = 0$ and $q = 1$ and $W_1 = 1$ thus

$$a_j = \alpha_j A_j(x)$$

and

$$F(x) = \underset{j=1,\cdots,n}{\text{Max}} a_j A_j(x)$$

which agrees with the approach suggested in [21]. We also note that if importance of A_j is one, $\alpha_j = 1$, then $a_j = A_j(x)$ and if all importances are one we get the form suggested earlier.

Conclusion

We have introduced a new class of operators that are useful for aggregating criteria guided by quantifiers.

References

[1] R. R. Yager, "Quantifiers in the formulation of multiple objective decision functions," *Inform. Sci.*, vol. 31, pp. 107–139, 1983.

[2] ____, "Aggregating evidence using quantified statements," *Inform. Sci.*, vol. 36, pp. 179–206, 1985.

[3] ____, "On the theory of bags," *Int. J. Gen. Syst.*, vol. 13, pp. 23–37, 1986.

[4] D. Dubois, "Triangular norms for fuzzy sets," in *Proc. 2nd Int. Seminar on Fuzzy Set Theory*, Linz, Austria, 1980, pp. 39–68.

[5] E. P. Klement, "Construction of fuzzy *g*-algebras using triangular norms," *J. Math. Anal. Appl.*, vol. 85, pp. 543–566, 1982.

[6] K. Menger, "Statistical metrics," *Proc. Nat. Acad. Sci. U.S.A.*, vol. 28, pp. 535–537, 1942.

[7] R. R. Yager, "On a general class of fuzzy connectives," *Fuzzy Sets and Syst.*, vol. 4, pp. 235–242, 1980.

[8] P. P. Bonissone and K. S. Decker, "Selecting uncertainty calculi and granularity: An experiment in trading off precision and complexity," in *Uncertainty in Artificial Intelligence*, L. N. Kanal and J. F. Lemmer, Eds. pp. 217–248. Amsterdam: North-Holland, 1986.

[9] R. R. Yager, "Some procedures for selecting fuzzy set-theoretic operators," *Int. J. Gen. Syst.*, vol. 8, pp. 115–124, 1982.

[10] L. A. Zadeh, "A computational approach to fuzzy quantifiers in natural languages," *Comps. & Maths. with Appls.*, vol. 9, pp. 149–184, 1983.

[11] ____, "A computational theory of dispositions," *Int. J. Intelligent Syst.*, vol. 2, pp. 39–64, 1987.

[12] R. R. Yager, "Quantified propositions in a linguistic logic," *Int. J. Man-Machine Studies*, vol. 19, pp. 195–227, 1983.

[13] L. A. Zadeh, "A theory of approximate reasoning," in *Machine Intell.*, vol. 9, J. E. Hayes, D. Michie and L. I. Kulich, pp. 149–194. New York: Wiley, 1979.

[14] A. DeLuca and S. Termini, "A definition of non-probabilistic entropy the setting of fuzzy sets theory," *Inform.*, vol. 20, pp. 301–312, 1972.

[15] D. Dubois and H. Prade, "Fuzzy set aggregation connectives," *Inform. Sci.*, vols. 1–2, pp. 85–122, 1985.

[16] H. Dyckhoff and W. Pedrycz, "Generalized means as models of comprehensive connectives," *Fuzzy Sets and Syst.*, vol. 14, pp. 143–154, 1984.

[17] R. R. Yager, "Fuzzy decision making including unequal objectives," *Fuzzy Sets and Syst.*, vol. 1, pp. 87–89, 1978.

[18] ____, "A linguistic variable for importance," *J. Cybern.*, vol. 10, pp. 249–260, 1980.

[19] ____, "A new methodology for ordinal multiple aspect decision making based on fuzzy sets," *Decision Sci.*, vol. 12, pp. 589–600, 1981.

[20] ____, "General multiple objective decision making and linguistically quantified statements," *Int. J. Man-Machine Studies*, vol. 21, pp. 389–400, 1984.

[21] ____, "A note on weighted queries in information retrieval systems," *J. Amer. Soc. Inform. Sci.*, vol. 38, pp. 23–24, 1987.

Correction to "Learning Optimal Discriminant Functions Through a Cooperative Game of Automata"

MANDAYAM A. L. THATHACHAR, SENIOR MEMBER, IEEE, AND P. S. SASTRY, MEMBER, IEEE

In the above paper,[1] equation (13) should have read

$$\hat{E}^n_{i_n}(k+1) = \max\left\{\hat{E}^n_{i_n}(k), \hat{d}_{i_1} \cdots i_N(k+1)\right\}, \quad \text{if } \hat{E}^n_{i_n}(k) \neq \hat{d}_{i_1} \cdots i_N(k)$$

$$= \max_{j_s, s \neq n} \left\{\hat{d}_{j_1} \cdots j_{n-1} i_n j_{n+1} \cdots j_N(k+1)\right\}, \quad \text{otherwise,}$$

$$\hat{E}^n_j(k+1) = \hat{E}^n_j(k), \qquad j \neq i_n.$$

Manuscript received November 23, 1987.

The authors are with Department of Electrical Engineering, Indian Institute of Science, Bangalore, 560012, India.

IEEE Log Number 8719231.

[1] M. A. L. Thathachar and P. S. Sastry, IEEE Trans. Syst., Man, Cybern., vol. SMC-17, no. 1, pp. 73–85, Jan./Feb. 1987.

0018-9472/88/0100-0190$01.00 ©1988 IEEE