ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

International Journal of Approximate Reasoning
48 (2008) 263–274

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

www.elsevier.com/locate/ijar

Prioritized aggregation operators

Ronald R. Yager

Machine Intelligence Institute, Iona College, New Rochelle, NY 10801, United States

Received 23 August 2007; accepted 29 August 2007
Available online 12 September 2007

Abstract

We consider criteria aggregation problems where there exists a prioritization relationship over the criteria. We suggest that prioritization between criteria can be modeled by making the weights associated with a criteria dependent upon the satisfaction of the higher priority criteria. We consider a number of aggregation operators in which there exists a prioritization relationship between the arguments. We first introduce a prioritized scoring operator and a closely related prioritized averaging operator. We next introduce a prioritized "anding" and then a prioritized "oring" operator.

1. Introduction

Many applications involve the selection or ordering of a group of alternatives based upon their satisfaction to a collection of criteria. Typical examples of this are information retrieval, multi-criteria decision making and database retrieval. Search engines such as Google require the solution of this type of problem. Central to the solution of these problems is the task of aggregation [1].

In these problems we have a collection of criteria $C = \{C_1, \ldots, C_n\}$ and a set of alternatives $X = \{x_1, \ldots, x_m\}$. We further have a measure of the satisfaction of criteria C_i by each alternative, $C_i(x) \in [0, 1]$. One commonly used approach is to calculate for each alternative x a score $C(x)$ as an aggregation of the $C_i(x)$

$$C(x) = F(C_1(x), \ldots, C_n(x))$$

and then order the alternatives using these scores. The form for F depends upon the users desired imperative for performing this aggregation. In addition in many types of applications one associates importance weights with the criteria [2–7]. A commonly used form for F is a weighted average of the $C_i(x)$. In this case we calculate

$$C(x) = \sum_{i=1}^{n} w_i C_i(x),$$

where the weights satisfy $w_i \in [0, 1]$ and $\sum_{i=1}^{n} w_i = 1$.

E-mail address: yager@panix.com

0888-613X/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2007.08.009

It is easy to see that this type of aggregation is monotonic in the sense that $C(x)$ does not decrease if any of the $C_i(x)$ increases. It is also bounded, $\text{Min}_i[C_i(x)] \leqslant C(x) \leqslant \text{Max}_i[C_i(x)]$. It is also idempotent, if all $C_i(x) = a$ then $C(x) = a$. Because of these properties this is an averaging operator. Closely related to this what we shall call a scoring (or more precisely a weighted scoring) operator. The difference between a scoring operator and a averaging operator is that the scoring operator does not require that $\sum_{i=1}^{n} w_i = 1$. We note that while a scoring operator is monotonic it is not necessarily bounded nor idempotent. Essentially a averaging operator is a special case of scoring operator. Both these operators can be used in the alternative selection problem.

Central to this types of aggregation operators is the ability to trade off between criteria. In particular $\frac{W_k}{Wi}$ is the relation between criteria C_i and C_k. In this type of aggregation we can compensate for a decrease of Δ in satisfaction to criteria C_i by gain $\frac{W_k}{Wi}\,\Delta$ in satisfaction to criteria C_k.

In many real applications we do not want to allow this kind of compensation between criteria. Consider the situation in which we are selecting a bicycle for our child based upon the criteria of safety and cost. In this situation we may not allow a benefit with respect to cost to compensate for a loss in safety. Here we have a kind of *prioritization* of the criteria. Safety has a higher priority. Consider a problem of document retrieval in which we are looking for documents about the American revolution and prefer if they are from an academic website and written after 2003. In this case the condition of it being about the American revolution has a priority, if it is not about this topic we are not interested. In organizational decision making criteria desired by superiors generally have a higher priority then those of their subordinates.

In this work we shall suggest aggregation operators that allows for the inclusion of priority between the criteria. Central to our approach will be the modeling of priority by using a kind of importance weight in which the importance of a lower priority criteria will be based on its satisfaction to the higher priority criteria [8]. As we shall see this result in a situation in which importance weights will not be the same across the alternatives.

2. Prioritized scoring and averaging operators

In the following we assume that we have a collection of criteria partitioned into q distinct categories, $H_1, H_2, \ldots, H_q$ such that $H_i = \{C_{i1}, C_{i2}, \ldots, C_{in_i}\}$, Here C_{ij} are the criteria in category H_i. We assume a prioritization between these categories

$$H_1 > H_2 > \cdots > H_q.$$

The criteria in the class H_i have a higher priority than those in H_k if $i < k$. The total set of criteria is $C = \cup_{i=1}^{q} H_i$. We assume $n = \sum_{i=1}^{q} n_i$ the total number of criteria.

In Fig. 1 we show the positioning of the criteria.

We assume that for any alternative $x \in X$ we have for each criteria C_{ij}, a value $C_{ij}(x) \in [0, 1]$ indicating its satisfaction to criteria C_{ij}.

In the following we introduce an aggregation operator F: $[0,1]^n \to [0,1]$ such that $F((a_{11}, \ldots, a_{1n_1}), \ldots, (a_{q1}, \ldots, a_{qn_q})) = \sum_{i=1}^{q} \left(\sum_{j=1}^{n_i} w_{ij} a_{ij} \right)$. We shall refer to as the prioritized scoring (PS) operator. This aggregation operator allows us to calculate $C(x)$ for any alternative as

$$C(x) = F(C_{ij}(x)) = \sum_{i=1}^{q} \left(\sum_{j=1}^{n_i} w_{ij} C_{ij}(x) \right).$$

Here the weights will also be a function of x and will be used to reflect the priority relationship. In order to obtain the weights for a given alternative x we proceed as follows.

For each priority category H_i we calculate

$$S_i = \text{Min}_j[C_{ij}(x)].$$

Here S_i is the value of the least satisfied criteria in category H_i under alternative x. Using this we will associate with each criteria C_{ij} a value u_{ij}. In particular for those criteria in category H_1 we have $u_{1j} = 1$. For those criteria in category H_2 we have $u_{2j} = S_1$. For those criteria in category H_2 we have $u_{3j} = S_1 S_2$. For those criteria in category H_4 we have $u_{4j} = S_1 S_2 S_3$. More generally u_{ij} is the product of the least satisfied criteria in all categories with higher priority than H_i.

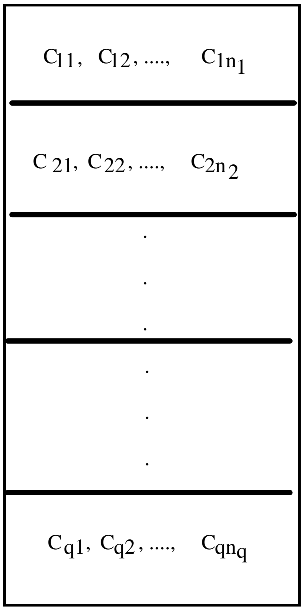

Fig. 1. Prioritization of criteria.

We can more succinctly and more generally express $u_{ij} = T_i$ where

$$T_i = \prod_{k=1}^{j} S_{k-1}$$

with the understanding that $S_0 = 1$ by default. We note that we can also express T_i as

$$T_i = S_{i-1}T_{i-1}.$$

This equation along with the fact that $T_1 = S_0 = 1$ gives a recursive definition at T_i.

We now see that for all $C_{ij} \in H_i$ we have $u_{ij} = T_i$. Using this we obtain for each C_{ij} a weight w_{ij} with respect to alternative x such that $w_{ij} = u_{ij}$. We see that each $w_{ij} \in [0, 1]$. We further observe that $T_i \geqslant T_k$ for $i < k$. From this it follows that if $i \leqslant j$ then $w_{ij} \geqslant w_{ke}$ for all j and e.

Using these weights we then can get an aggregated score x under these prioritized criteria as

$$C(x) = \sum_{i,j} w_{ij}C_{ij}(x) = \sum_{i,j} T_i C_{ij}(x) = \sum_{i=1}^{q} T_i \left(\sum_{j=1}^{n_i} C_{ij}(x)\right).$$

We note that this operator is monotonic, if $C_{kj}(x)$ increases then $C(x)$ cannot decrease. We see this as follows: $\frac{\partial C(x)}{\partial C_{kj}(x)} = T_k + \sum_{i=k+1}^{q} \frac{\partial T_i}{\partial C_{kj}(x)} \left(\sum_{j=1}^{n_i} C_{ij}(x)\right)$. If $S_k \neq C_{kj}(x)$ then $\frac{\partial T_i}{\partial C_{kj}(x)} = 0$ for $i \geqslant k+1$ and hence $\frac{\partial C(x)}{\partial C_{kj}(x)} = T_k \geqslant 0$. If $S_k = C_{kj}(x)$ then for $i \geqslant k+1$ we have $\frac{\partial T_i}{\partial C_{kj}(x)} = \prod_{r=1 \text{ to } i-1, r\neq k}^{j-1} S_r \geqslant 0$ and hence again $\frac{\partial C(x)}{\partial C_{kj}(x)} \geqslant 0$. Following is an example using this PS operator.

Example: Consider the following prioritized collection of criteria:

$H_1 = \{C_{11}, C_{12}\}$,
$H_2 = \{C_{21}\}$,
$H_3 = \{C_{31}, C_{32}, C_{33}\}$,
$H_4 = \{C_{41}, C_{42}\}$.

Assume for alternative x we have

$C_{11}(x) = 0.7,\ C_{12}(x) = 1,$
$C_{21}(x) = 0.9,$

$C_{31}(x) = 0.8,\ C_{32}(x) = 1,\ C_{33}(x) = 0.2,$
$C_{41}(x) = 1,\ C_{42}(x) = 0.9.$

We first calculate

$S_1 = \mathrm{Min}[C_{11}(x), C_{12}(x)] = 0.7,$
$S_2 = \mathrm{Min}[C_{21}(x)] = 0.9,$
$S_3 = \mathrm{Min}[C_{31}(x), C_{32}(x), C_{33}(x)] = 0.2,$
$S_4 = \mathrm{Min}[C_{41}(x), C_{42}(x)] = 0.9.$

Using this we get

$T_1 = 1,$
$T_2 = S_1T_1 = 0.7,$
$T_3 = S_2T_2 = 0.63,$
$T_4 = S_3T_3 = 0.12.$

From this we obtain

$u_{11} = u_{12} = T_1 = 1,$
$u_{21} = T_2 = 0.7,$
$u_{31} = u_{32} = u_{33} = T_3 = 0.63,$
$u_{41} = u_{42} = T_4 = 0.12.$

In this case then we have

$w_{11} = w_{12} = 1,$
$w_{21} = 0.7,$
$w_{31} = w_{32} = w_{33} = 0.63,$
$w_{41} = w_{42} = 0.12.$

We now calculate $C(x) = \sum_{ij} w_{ij}C_{ij} = 3.82$.

We now look at some further properties of the proposed aggregation method. We recall $H_i = \{C_{ij}|j = 1\text{–}n_i\}$ where the criteria in category H_i have priority over those in H_k if $i_i i < k$. Again letting $a_{ij} = C_{ij}(x)$ we have $S_i = \mathrm{Min}_j[a_{ij}]$ and $S_o = 1$ and $T_i = \prod_{k=1}^{j} S_{k-1}$. Here with $u_{ij} = T_i$ we use as our weights in this prioritized scoring operator $w_{ij} = u_{ij} = T_i$ and hence

$$C(x) = \sum_{i=1}^{q}\left(\sum_{j=1}^{n_i} w_{ij}a_{ij}\right) = \sum_{i=1}^{q} T_i\left(\sum_{j=1}^{n_i} a_{ij}\right).$$

Letting $A_i = \sum_{j=1}^{n_i} a_{ij}$ we have $C(x) = \sum_{i=1}^{q} T_iA_i$.

We see that the weight associated with the elements in the ith category is $T_i = \Pi_{k=1}{}^{i}S_{k-1}$. Thus the criteria in H_i contribute proportionally to the product of the satisfaction of the higher order criteria. Thus poor satisfaction to any higher criteria reduces the ability for compensation by lower priority criteria. This is of course the fundamental feature of the prioritization relationship.

We also observe that if there exists some category H_r such that $C_{rj}(x) = 0$ for some criteria in H_r then $S_r = 0$ and $T_i = 0$ for $i > r$ and hence $C(x) = \sum_{i=1}^{r} T_iA_i$.

Note: While in the preceding we assumed $C_{ij}(x) \in [0, 1]$ this is not necessarily required. If we let $F_{ij} : \mathscr{R} \to [0, 1]$ be some function from the real numbers into the unit intervals such that $F_{ij}(C_{ij}(x))$ is some measure of how satisfied we are with a score $C_{ij}(x)$ for criteria C_{ij} then we allow the values of $C_{ij}(x)$ be any number if we calculate

$S_i = \text{Min}_j[F_{ij}(C_{ij}(x))]$.

Here we just transfer the $C_{ij}(x)$ into numbers in the unit interval for calculating S_i.

A natural question that arises is why have we chosen this scoring type operator rather then an averaging operator which requires that the $\sum_{ij} w_{ij} = 1$. We see this can be easily accomplished by a simple normalization. In particular if instead of using $w_{ij} = u_{ij}$ we use $w_{ij} = \frac{u_{ij}}{\sum_{i=1}^{q}\sum_{j=1}^{n_i} u_{ij}}$ since $u_{ij} = T_i$. This simplifies to $w_{ij} = \frac{T_i}{\sum_{i=1}^{q} n_i T_i}$. As the following example illustrates performing this normalization does not always guarantee a monotonic aggregation.

Example: Assume $H_1 = \{C_{11}, C_{12}, C_{13}, C_{14}\}$ and $H_2 = \{C_{21}, C_{22}, C_{23}\}$. Assume for x we have $C_{11}(x) = C_{12}(x) = C_{13}(x) = 1$, $C_{14}(x) = 0$ and $C_{21}(x) = C_{22}(x) = C_{23}(x) = 0$. In this case $S_1 = 0$ and hence $T_1 = 1$ and $T_2 = 0$. Thus we get $u_{1j} = 1$ and $u_{2j} = 0$ and hence $\sum_{i=1}^{q}\sum_{j=1}^{n_i} u_{ij} = 4$. From this we get $w_{1j} = \frac{1}{4}$ for $j = 1$–4 and $w_{2j} = 0$ for $j = 1$–3 and therefore

$$C(x) = \frac{1}{4}(C_{11}(x) + C_{12}(x) + C_{13}(x) + C_{14}(x)) = 0.75.$$

Consider alternative y for which we have $C_{11}(y) = C_{12}(y) = C_{13}(y) = 1$, $C_{14}(y) = 1$ and $C_{21}(y) = C_{22}(y) = C_{23}(y) = 0$. The only difference between x and y is that we have increased the satisfaction of C_{14}, $C_{14}(y) = 1$ while $C_{14}(x) = 0$. Monotonicity requires that $C(y) \geqslant C(x)$. Let us calculate $C(y)$. In this case $S_1 = 1$ and therefore $T_1 = 1$ and $T_2 = 1$. In this case all $u_{ij} = 1$ and hence $\sum_{ij} u_{ij} = 7$ and therefore all $w_{ij} = \frac{1}{7}$. From this we get that

$$C(y) = \frac{1}{7}\sum_{ij} C_{ij}(y) = \frac{4}{7} = 0.57 < 0.75.$$

Thus we see that $C(y) < C(x)$ and the monotonicity condition has not been satisfied.

We note the use of a scoring type aggregation operator does indeed respect the monotonicity. In this case $w_{ij} = u_{ij}$. Hence for x we have $w_{1j} = u_{1j} = 1$ and $w_{2j} = u_{2j} = 0$ From this we get $C(x) = 3$. For the case of y we get $w_{1j} = u_{1j} = 1$ and $w_{2j} = u_{2j} = 1$. From this we get $C(y) = 4$ and hence the monotonicity is respected.

In the preceding the priority relationship between the criteria was a weak ordering, we allowed ties as was the case for criteria in the same category. As we shall subsequently show if the priority relationship between the criteria is a linear ordering, no ties allowed, then we can obtain a prioritized averaging (PA) operator.

He we also assume we have a collection of criteria partitioned into q distinct categories, $H_1, H_2, \ldots, H_q$ and we assume a prioritization between these categories $H_1 > H_2 > \cdots > H_q$. However here we assume each category has just one member $H_i = \{C_i\}$. Thus here there is a linear ordering among the criteria $C_1 > C_2 > \cdots > C_q$. We have used only one index as we have no need for the second index. Our objective is to get a collection of weights w_i that respect the prioritization and use these to calculate $C(x) = \sum_{i=1}^{q} w_j C_j(x)$ Since we want this to be a prioritized averaging operator we require that $w_i \in [0, 1]$ and $\sum_{i=1}^{q} w_i = 1$. In order to obtain these weights we shall essentially follow the procedure used in the preceding with the addition of a normalization step.

For each priority category H_i we calculate S_i as the value of the least satisfied criteria in H_i, in this case we simply get $S_i = C_i(x)$. Again here we let $T_1 = 1$ and for $i > 1$ we let $T_i = \prod_{k=1}^{i-1} S_k$. If we let $S_0 = 1$ we can more succinctly express this as $T_i = \prod_{k=1}^{i} S_{k-1}$ for all i. Denoting $u_i = T_i$ as the un-normalized weights we can obtain normalized weights $w_i = \frac{u_i}{T}$ where $T = \sum_{i=1}^{q} u_i = \sum_{i=1}^{q} T_i$.

It is clear that the w_i lie in the unit interval and sum to one. To assure that $C(x) = \sum_{i=1}^{q} w_i C_i(x)$ is an averaging operator we must show that it is bounded and monotonic.

We now show that this aggregation method is bounded and monotonic. First we see that the value of this aggregation is bounded by the maximum and minimum of the arguments and hence it is also idempotent. For simplicity let us denote $a_i = C_i(x)$. Using this we have $C(x) = \sum_{i=1}^{q} w_i a_i$.

Consider now boundedness. Assume $a = \text{Min}_i[a_i]$ and $b = \text{Max}_i[a_j]$ then $C(x) = \sum_{i=1}^{q} w_i a_i \geqslant a$ and $C(x) = \sum_{i=1}^{q} w_i a_i \leqslant b$. Now consider the case where all the a_i are the same, $a_i = d$. In this case since $\sum_{i=1}^{q} w_i = 1$ we get $C(x) = \sum_{i=1}^{q} w_i d = d$ and hence the operation is idempotent.

We now consider the issue of monotonicity. We shall denote the satisfaction of each criteria to x as $a_i = C_i(x)$. We note that in this case with one criteria at each level, $S_i = a_i$. Here then $T_1 = 1$, $T_2 = a_1$ and more generally $T_i = \prod_{k=1}^{i} S_{k-1}$. Using this we have

$$C(x) = \frac{\sum_{i=1}^{q} T_i a_i}{T}.$$

Let us denote $C(x) = M/T$ where $M = \sum_{i=1}^{q} T_i a_i$ and $T = \sum_{i=1}^{q} T_i$. For monotonicity to hold we have to show that $\frac{\partial C(x)}{\partial a_j} \geqslant 0$ for any j. This requires that

$$\frac{T\frac{\partial M}{\partial a_j} - M\frac{\partial T}{\partial a_j}}{T^2} \geqslant 0.$$

Hence we must show that the numerator is non-negative,

$$T\frac{\partial M}{\partial a_j} - M\frac{\partial T}{\partial a_j} \geqslant 0.$$

Before preceding we note that $\frac{\partial T_i}{\partial a_j} = 0$ for $i \leqslant j$ and $\frac{\partial T_i}{\partial a_j} = \frac{T_i}{a_j}$ for $i > j$. We also note that $M = \sum_{i=1}^{q} T_i a_i = \sum_{i=1}^{q} T_{i+1}$ since $T_i\, a_i = T_{i+1}$. However we shall find it more useful to express $M = \sum_{i=2}^{q+1} T_i$.

We shall denote $A = \frac{\partial M}{\partial a_j} = \frac{1}{a_j}\sum_{i=j+1}^{q+1} T_i$. We shall also let $B = \frac{\partial T}{\partial a_j}$ hence since $T = \sum_{i=1}^{q} T_i$ we have $B = \frac{\partial T}{\partial a_j} = \frac{1}{a_j}\sum_{i=j+1}^{q+1} T_i$. From this we observe that $A \geqslant B$. In the following we shall find it convenient to denote $E = \sum_{i=2}^{j} T_i$.

Consider now the term $T\frac{\partial M}{\partial a_j} - M\frac{\partial T}{\partial a_j} = AT - BM$. We now observe that

$$T = \sum_{i=1}^{q} T_i = \sum_{i=1}^{j} T_i + a_j B.$$

Since $T_1 = 1$ then $T = 1 + E + a_j B$. We further observe that

$$M = \sum_{i=2}^{q+1} T_i. = E + a_j A.$$

Using the relations we see that

$$AT - BM = A(1 + E + a_j B) - B(E + a_j A) = A + EA + a_j BA - BE - a_j BA,$$
$$AT - BM = A + E(A - B).$$

Since $A \geqslant B$ it follows that $AT - BM \geqslant 0$.

Thus for linear ordered criteria we can obtain a prioritized averaging aggregation operator.

3. Alternative determination of weights

In the preceding we introduced the prioritized scoring operator as a method for multi-criteria aggregation for the case in which our criteria where partitioned into q categories, $H_i = \{C_{ij} : j = 1, \ldots, n_i\}$ where category H_i had priority over H_k if $i < k$. For a given alternative x we shall find it convenient in the following to denote $C_{ij}(x) = a_{ij}$. Using this notation then we defined

$$S_o = 1,$$
$$S_i = \text{Min}_i[a_{ij}] \quad \text{for } i = 1 \text{ to } q,$$
$$T_i = \prod_{k=1}^{j} S_{k-1} \quad \text{for } i = 1 \text{ to } q.$$

With $w_{ij} = T_i$ we obtained as our aggregated value

$$C(x) = \sum_{i=1}^{q}\sum_{j=1}^{n_i} w_{ij} a_{ij} = \sum_{i=1}^{q}\sum_{j=1}^{n_i} T_i a_{ij}.$$

Letting $A_i = \sum_{j=1}^{n_i} a_{ij}$ we can express this as $C(x) = \sum_{i=1}^{n} T_i A_i$.

In the preceding we assumed that the satisfaction to the priority class $H_i = \{C_{i1}, \ldots, C_{in_i}\}$ under alternative x was determined by the least satisfied criteria in H_i, $S_i = \text{Min}_j[C_{ij}(x)]$. Here we shall suggest some alternative methods for calculating S_i.

One method we shall consider will be based on the OWA aggregation operator [9,10]. Here we associate with each priority class H_i a vector $\mathbf{V}_i$ of dimension n_i called the OWA weighting vector. The components V_{ik} of $\mathbf{V}_i$ are such that $V_{ik} \in [0,1]$ and $\sum_{k=1}^{n_i} V_{ik} = 1$. Additionally we let $\text{ind}_i(k)$ be an index of function so that $b_{ik}(x) = C_{ind_i(k)}(x)$ is the kth largest of $C_{ij}(x)$. Using this we now calculate

$$S_i = \sum_{k=1}^{n_i} V_{ik} b_{ik}(x).$$

We see that if $V_{in_i} = 1$ and $V_{ik} = 0$ for $k \neq n_i$ then we get $S_i = \text{Min}_j[C_{ij}(x)]$, the original method. An important special case is where $V_{ik} = 1/n_i$ for all k. In this case $S_i = \frac{1}{n_i}\sum_{j=1}^{n_i} C_{ij}(x)$. Here we take as S_i the average of the satisfactions of the criteria in category H_i. Another special case is when $V_{i1} = 1$ and $V_{ik} = 0$ for $k \neq 1$. In this case $S_i = \text{Max}_j[C_{ij}(x)]$. Here we take S_i as the score of the most satisfied criteria in category H_i. Many other weight vectors are possible for example if $V_{iq} = 1$ for some qS_i simply becomes the qth largest of the $C_{ij}(x)$.

In this framework we can associate with each weighing vector $\mathbf{V}_i$ a measure called its attitudinal character denoted, A-C($\mathbf{V}_i$) [11]. We define this as

$$\text{A-C}(\mathbf{V}_i) = \frac{1}{n_i - 1}\sum_{k=1}^{n_i} V_{ik}(n_i - k).$$

It can easily be shown [9] that for the case where $V_{ini} = 1$ we get A-C($\mathbf{V}_i$) = 0. For the case where $V_{ik} = 1/n_i$ for all k then A-C($\mathbf{V}_i$) = 0.5 and for the case where $V_{i1} = 1$ we have A-C($\mathbf{V}_i$) = 1.

If we denote A-C(V_c) = α_i then we see in Fig. 2 the relationship between the value of α_i and the form for the calculation of S_i. Here then α_ι can be seen as a measure of the *tolerance* in determining the satisfaction of the category. While it is not necessary, it would be seen that the default situation is to assume α_ι is the same for all H_i.

Many of the techniques available for calculating the OWA weights [12] can be tailored for this particular application. A particularly interesting possibility is to use a variation of the method originally suggested by O'Hagan [13–15]. In this case we would supply a desired level of tolerance α_i and solve the following mathematical programming problem for the V_{ik}:

$$\begin{aligned}
&\text{Min} && \sum_{k=1}^{n_i}(V_{ik})^2 \\
&\text{Such that :} && \frac{1}{n_i - 1}\sum_{k=1}^{n_i} V_{ik}(n_i - k) = \alpha_i, \\
& && \sum_{k=1}^{n_i} V_{ik} = 1, \\
& && V_{ik} \geqslant 0.
\end{aligned}$$

We provide an example of the preceding variation using the earlier example

Example: $H_1 = \{C_{11}, C_{12}\}$, $H_2 = \{C_{21}\}$, $H_3 = \{C_{31}, C_{32}, C_{33}\}$, $H_4 = \{C_{41}, C_{42}\}$

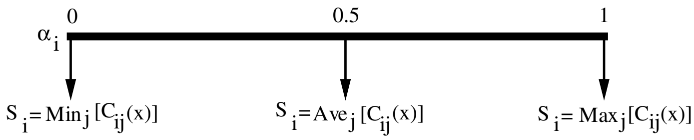

Fig. 2. Relationship between α_i and the form of S_i.

For alternative x we have

$C_{11}(x) = 0.7,\ C_{12}(x) = 1,$
$C_{21}(x) = 0.9,$
$C_{31}(x) = 0.8,\ C_{32}(x) = 1,\ C_{33}(x) = 0.2,$
$C_{41}(x) = 1,\ C_{42}(x) = 0.9.$

Consider the case where $S_i = \text{Max}_j[C_{ij}(x)]$. Here then

$$S_1 = 1, \quad S_2 = 0.9, \quad S_3 = 1, \quad S_4 = 1.$$

From this we get

$$T_1 = 1, \quad T_2 = S_1T_1 = 1, \quad T_3 = S_2T_2 = 0.9, \quad T_4 = S_3T_3 = 0.9.$$

With $C(x) = \sum_{i=1}^{4} A_iT_i$ where $A_i = \sum_{j=1}^{n_i} C_{ij}(x)$ we have

$$C(x) = (1)(1.7) + (1)(0.9) + (0.9)(2) + (0.9)(1.9) = 6.11.$$

Another approach for calculating the S_i involves associating with each criteria in H_i an additional local weight. In this case our form for H_i is

$$H_i = \{(C_{ij}, g_{ij})|j = 1, \ldots, n_i\},$$

where the g_{ij} indicates the importance of C_{ij} in calculating S_i. Here we assume that $g_{ij} \in [0, 1]$ and $\sum_{j=1}^{n_i} g_{ij} = 1$. Using these weights we can calculate $S_i = \sum_{j=1}^{n_i} g_{ij}C_{ij}(x)$.

An interesting special case of this is where some criteria C_{ij} has $g_{ij} = 0$. In this case the criteria plays no role in the determination of S_i but still is able to contribute to the overall calculation of $C(x)$.

Another available method for calculating the S_i involves the idea of combining these local weights with a tolerance level. Here we assume for each H_i we have $H_i = \{(C_{ij}, g_{ij}), j = 1, \ldots, n_i\}$, $g_{ij} \in [0, 1]$ and $\sum_{j=1}^{n_i} g_{ij} = 1$, where again g_{ij} is the indication at the importance of C_{ij} in calculating S_i. In addition we assume a tolerance level $\alpha_i \in [0, 1]$ associated with H_i. Using one of the methods for generating OWA weights we can obtain a set of OWA weights, V_{ik}, for $k = 1$–n_i. Let nd_i be an index such $nd_i(k)$ is the index of the k largest of the $C_{ij}(x)$. That is $b_{ik} = C_{i,nd_i(k)}(x)$ is the value of the k most satisfied criteria in H_i. With $d_{ik} = g_{i,nd_i(k)}$ being the importance weight associated with this kth most satisfied criteria on H_i we calculate

$$h_{ik} = \frac{d_{ik}V_{ik}}{\sum_{k=1}^{n_i} d_{ik}V_{ik}}.$$

Using this we calculate

$$S_i = \sum_{k=1}^{n_i} h_{ik}b_{ik}.$$

In the special case when $V_{ik} = 1/n_i$ for all k this reduces to the weighted average introduced earlier, $S_i = \sum_{j=1}^{n_i} g_{ij}C_{ij}(x)$.

4. Prioritized "and" operator

In the following we shall consider a related aggregation method called prioritized *anding*. We refer to this as the PRI-AND aggregation operator.

We recall the "and" operator is generalized by a t-norm [15,17]. A t-norm[1] is a mapping.

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

having the properties

[1] In the following I use the notation 9i instead the standard notation of T for the t-norm as I used T for something else, I hope this non-standard notation is not a problem.

1. *Symmetry*: $R(a,b) = R(b,a)$;
2. *Monotonicity*: If $a \geqslant c$ and $b \geqslant d$ then $R(a,b) \geqslant R(c,d)$;
3. *Associativity*: $R(a, R(b,c)) = R(R(a,b),c)$;
4. *1 as identity*: $R(1,a) = a$.

The associativity property allows us to extend this to any number of arguments. An interesting property of the t-norm is $R(a_1,\ldots,a_n) \geqslant R(a_1,\ldots,a_n,a_{n+1})$.

A large number of possible examples of t-norms exist [18]. Three of the most important are

$R_M(x,y) = \text{Min}(x,y)$ Minimum
$R_P(x,y) = xy$ Product
$R_L(x,y) = \text{Max}(x+y-1,0)$ Lukasiewicz

It can be shown that for any $x, y, R_M(x,y) \geqslant R_P(x,y) \geqslant R_L(x,y)$. It is also true that for any t-norm R it is the case that $R_M(x,y) \geqslant R(x,y)$.

We now look at the issue of performing the t-norm aggregation when the arguments have importance weights associated with them [3,19]. Consider the aggregation $R((a_1,w_1),(a_2,w_2),\ldots,(a_n,w_n))$ where $w_j \in [0,1]$ is the importance weight associated with the argument a_j. In [2], Yager suggested that we can implement this aggregation as

$$R((a_1,w_1),\ldots,(a_n,w_n)) = R(a_i^{w_i},\ldots,a_n^{w_n}).$$

For example in the case where $R = R_P$ we have

$$R((a_1,w_1),\ldots(a_n,w_n)) = \prod_{i=1}^{n} a_i^{w_i}.$$

In the case where $R = R_M$ then

$$R((a_1,w_1),\ldots,(a_n,w_n)) = \text{Min}_i[a_i^{w_i}].$$

We note that if $w_i = 0$ then $a_i^0 = 1$. Since one is the identity of the t-norm then criteria with zero importance have no effect in the calculations of $R(a_i^{w_i},\ldots,a_n^{w_n})$.

In [20], Yager suggested alternate methods for implementing weighted t-norm aggregations. While we shall not discuss these here we do note that the methodology introduced in the following can be easily applied to any of the other methods for including importance.

In the following we introduce a prioritized 'and' operator. We refer to this as the PRI-AND aggregation operator.

Again we have a collection of criteria partitioned into q categories $\{H_1,\ldots,H_q\}$ such that $H_i = \{C_{i1},\ldots,C_{in_i}\}$. Again we assume a prioritization of the categories, $H_1 > H_2 > \cdots > H_q$. Our objective is to obtain a prioritized 'anding' aggregation of the satisfaction of these criteria by some alternative x. We assume $C_{ij}(x) \in [0,1]$ is the satisfaction of criteria C_{ij} by alternative x.

We first calculate for each category $S_i = \text{Min}_j[C_{ij}(x)]$. We next calculate

$$T_i = \prod_{k=1}^{i} S_{k-1}$$

with the understanding that $S_o = 1$ by definition. We now define the prioritized weight associated with C_{ij} as $w_{ij} = T_i$. We now calculate the PRI-AND aggregation of the $C_{ij}(x)$ using the t-norm R as

$$C(x) = \boldsymbol{R}_{i,j}[(C_{ij}(x))^{w_{ij}}].$$

Since $w_{ij} = T_i$ then $C(x) = \boldsymbol{R}_i, \boldsymbol{R}_j[(C_{ij}(x))^{T_i}]$.

In the case where R is the Min then we get $C(x) = \text{Min}_i[\text{Min}_j(C_{ij(x)}^{\text{T}_i})]$

Noting that $\text{Min}_j(C_{ij}(x)) = S_i$ we have

$$C(x) = \text{Min}_i[S_i^{T_i}].$$

In the case where R is the product t-norm we have

$$C(x) = \prod_i \prod_j [(C_{ij}(x))^{T_i}]$$

If we get $D_i(x) = \prod_{j=1}^{n_i} C_{ij}(x)$ then we get

$$C(x) = \prod_{i=1}^{q} (D_i)^{T_i}.$$

If we take the log of the above we have

$$\mathrm{Log}(C(x)) = \sum_{i=1}^{q} T_i \mathrm{Log}(D_i).$$

Since $D_i(x) = \prod_{j=1}^{n_i} C_{ij}(x)$ then $\mathrm{Log}(D_i) = \sum_{j=1}^{n_i} \mathrm{Log}(C_{ij}(x))$. Hence in this case we have

$$\mathrm{Log}(C(x)) = \sum_{i=1}^{q} T_i \mathrm{Log}(D_i) \quad \text{and} \quad \mathrm{Log}(D_i) = \sum_{j=1}^{n_i} \mathrm{Log}(C_{ij}(x)).$$

This form looks very similar to the weighted average where

$$C(x) = \frac{1}{T} \sum_{i=1}^{q} T_i A \quad \text{and} \quad A_i = \sum_{j=1}^{n_i} C_{ij}(x).$$

5. Prioritized "or" operator

We now consider a related aggregation method the prioritized *oring*. We refer to this as PRI-OR aggregation operator.

We recall that the "or" operator is generalized by a t-conorm[2] [16], a mapping.

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

having the properties

1. *Symmetry*: $P(a,b) = P(b,a)$.
2. *Monotonicity*: If $a \geqslant c$ and $b \geqslant d$ then $P(a,b) \geqslant P(c,d)$.
3. *Associativity*: $P(a, P(b,c)) = P(P(a,b),c)$.
4. 0 *as identity*: $P(0,a) = a$.

A property of the t-conorm is $P(a_1,\ldots,a_n) \leqslant P(a_1,\ldots,a_n,a_{n+1})$.

Three important are examples of this are

$P_M(x,y) = \mathrm{Max}(x,y)$ Maximum
$P_S(x,y) = x + y - xy$ Probabilistic Sum
$P_L(x,y) = \mathrm{Min}(x+y,1)$ Lukasiewicz

It is well know that for any t-conorm P it is the case that $P_M(x,y) \leqslant P(x,y)$, max is the smallest.

We now look at the issue of performing the t-conorm aggregation when the arguments have importance weights associated with them. Consider the aggregation $P((a_1,w_1),(a_2,w_2),\ldots,(a_n,w_n))$ where $w_j \in [0,1]$ is the importance weight associated with the argument a_j. In [3], Yager suggested that we can implement this aggregation as

[2] In the following I use the notation P instead the standard notation of S for the t-conorm as I used S for something else, I hope this non-standard notation is not a problem.

$$P((a_1, w_1), \ldots, (a_n, w_n)) = P(w_1 a_1, w_2 a_2, \ldots, w_n a_n).$$

We aggregate the product of w_j times a_j. For example in the case where $P = P_M$ we have

$$P_M((a_1, w_1), \ldots, (a_n, w_n)) = \text{Max}_i[w_i a_i].$$

Consider now the case of probabilistic sum. Since $R_p(x \cdot y) = 1 - (1 - x)(1 - y)$ then

$$P_s((a_1, w_1), \ldots, (a_n, w_n)) = 1 - \prod_{i=1}^{n}(1 - w_i a_i).$$

We note that if $w_i = 0$ then $w_i a_i = 0$. Since zero is the identity of the t-conorm then criteria with zero importance have no effect in the calculations of $P(w_1 a_1, w_2 a_2, \ldots, w_n a_n)$.

In the following we introduce a prioritized 'or' operator, the PRI-OR aggregation operator.

Here we have a collection of criteria partitioned into q categories $\{H_1, \ldots, H_q\}$ such that $H_i = \{C_{i1}, \ldots C_{in_i}\}$. Again we assume a prioritization of the categories, $H_1 > H_2 > \cdots > H_q$. Our objective is to obtain a prioritized 'oring' aggregation of the satisfaction of these criteria by some alternative x. We assume $C_{ij}(x) \in [0, 1]$ is the satisfaction of criteria C_{ij} by alternative x.

We first calculate for each category $S_i = \text{Max}_j[C_{ij}(x)]$. We next calculate

$$T_i = \prod_{k=1}^{i} S_{k-1}$$

with the understanding that $S_o = 1$ by definition. We define the prioritized weight associated with C_{ij} as $w_{ij} = T_i$. We now calculate the PRI-OR aggregation of the $C_{ij}(x)$ using the t-conorm P as

$$C(x) = \boldsymbol{P}_{i,j}[(w_{ij} C_{ij}(x))].$$

Since $w_{ij} = T_i$ then $C(x) = \boldsymbol{P}_i \boldsymbol{P}_j[(T_i C_{ij}(x))]$.

To get a feel for this we consider the special case where each category has just one element, $H_i = \{C_i\}$ and P is the probabilistic sum. In this case $S_i = C_i(x)$ with $S_o = 1$. Furthermore $T_i = \prod_{k=1}^{i} S_{k-1} = \prod_{k=1}^{i-1} C_k(x)$. In this case

$$C(x) = 1 - \prod_{i=1}^{q}\left(1 - \left(C_i(x) \prod_{k=1}^{i-1} C_k(x)\right)\right) = 1 - \prod_{i=1}^{q}\left(1 - \prod_{k=1}^{i} C_k(x)\right).$$

For the case where $q = 2$ we have

$$C(x) = 1 - (1 - C_1(x))(1 - C_1(x)C_2(x)) = C_1(x) + C_1(x)C_2(x) - C_1(x)C_1(x)C_2(x),$$
$$C(x) = C_1(x)(1 + C_2(x)(1 - C_1(x))) = C_1(x)(1 + C_2(x)\overline{C_1(x)}).$$

Thus if $C_1(x) = 1$ then $C(x) = 1$ and if $C_1(x) = 0$ then $C(x) = 0$. If for example $C_1(x) = 0.7$ then

$$C(x) = 0.7(1 + 0.3C_2(x)).$$

On the other hand if $C_2(x) = 0$ then $C(x) = C_1(x)$ while if $C_2(x) = 1$ then

$$C(x) = C_1(x) + C_1(x)\overline{C_1(x)}.$$

6. Conclusion

We considered criteria aggregation problems where there is a prioritization relationship over the criteria. We suggested that prioritization between criteria can be modeled by making the weights associated with a criteria dependent upon the satisfaction of the higher priority criteria. This resulted in a situation in which the weights associated with the criteria depended upon the alternative being evaluated. We introduce a number of aggregation operators in which there exists a prioritization relationship between the arguments. We first introduced a prioritized scoring operator. We showed that in the special case where prioritization relationship among the criteria satisfies a linear ordering we can obtain a prioritized averaging operator. We next introduced a prioritized "anding" and then a prioritized "oring" operator.

References

[1] T. Calvo, A. Kolesarova, M. Komornikova, R. Mesiar, Aggregation operators: properties, classes and construction methods, in: T. Calvo, G. Mayor, R. Mesiar (Eds.), Aggregation operators, Physica-Verlag, Heidelberg, 2002, pp. 3–104.
[2] R.R. Yager, Multiple objective decision making using fuzzy sets, Int. J. Man–Machine Stud. 9 (1977) 375–382.
[3] R.R. Yager, A note on weighted queries in information retrieval systems, J. Am. Soc. Inform. Sci. 38 (1987) 223–224.
[4] V. Torra, The weighted OWA operator, Int. J. Intell. Syst. 12 (1997) 153–166.
[5] T. Calvo, R. Mesiar, Weighted means based on triangular conorms, Int. J. Uncertain. Fuzziness Knowledge Based Syst. 9 (2001) 183–196.
[6] V. Torra, Learning weights for quasi-weighted means, IEEE Trans. Fuzzy Syst. 10 (2002) 653–666.
[7] R. Mesiar, T. Calvo, R.R. Yager, Quantitative weights and aggregation, IEEE Trans. Fuzzy Syst. 12 (2004) 62–69.
[8] R.R. Yager, Modeling prioritized multi-criteria decision making, IEEE Trans. Syst., Man Cybern. Part B 34 (2004) 2396–2404.
[9] R.R. Yager, On ordered weighted averaging aggregation operators in multi-criteria decision making, IEEE Trans. Syst., Man Cybern. 18 (1988) 183–190.
[10] R.R. Yager, J. Kacprzyk, The Ordered Weighted Averaging Operators: Theory and Applications, Kluwer, Norwell, MA, 1997.
[11] R.R. Yager, On the cardinality index and attitudinal character of fuzzy measures, Int. J. Gen. Syst. 31 (2002) 303–329.
[12] Z. Xu, An overview of methods for determining OWA weights, Int. J. Intell. Syst. 20 (2005) 843–865.
[13] M. O'Hagan, A fuzzy neuron based upon maximum entropy-ordered weighted averaging, in: B. Bouchon-Meunier, R.R. Yager, L.A. Zadeh (Eds.), Uncertainty in Knowledge Bases, Springer-Verlag, Berlin, 1990, pp. 598–609.
[14] M. O'Hagan, Using maximum entropy-ordered weighted averaging to construct a fuzzy neuron, in: Proceedings 24th Annual IEEE Asilomar Conf. on Signals, Systems and Computers, Pacific Grove, CA, 1990, pp. 618–623.
[15] R. Fuller, P. Majlender, On obtaining minimal variability OWA weights, Fuzzy Sets Syst. 136 (2003) 203–215.
[16] C. Alsina, E. Trillas, L. Valverde, On some logical connectives for fuzzy set theory, J. Math. Anal. Appl. 93 (1983) 15–26.
[17] E.P. Klement, R. Mesiar, E. Pap, Triangular Norms, Kluwer Academic Publishers, Dordrecht, 2000.
[18] G.J. Klir, T.A. Folger, Fuzzy Sets, Uncertainty and Information, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[19] R.R. Yager, Weighted triangular norms using generating functions, Int. J. Intell. Syst. 19 (2004) 217–231.
[20] R.R. Yager, Structures generated from weighted fuzzy intersection and union, J. Chin. Fuzzy Syst. Assoc. 2 (1996) 37–58.