Multiple Priorities in an Induced Ordered Weighted Averaging Operator

H. B. Mitchell,* P. A. Schaefer†
Intelligence Centers Department, Elta Electronics Industries Ltd., Ashdod, Israel

The ordered weighted averaging (OWA) operator of Yager was introduced to provide a method for nonlinearly aggregating a set of input arguments a_i. A fundamental aspect of the OWA operator is a reordering step in which the input arguments are rearranged according to their values. Recently, an induced OWA operator was described in which each input argument a_i has a single priority α_i value. The input arguments are then rearranged according to the *priorities* α_i and not their own values. In this paper we show how the induced OWA operator may be defined when each argument a_i has associated with it N priority values α_{ip}, $p \in \{1, 2, \dots, N\}$, instead of a single priority value. Finally, we suggest two applications using a multiple priority induced OWA operator. © 2000 John Wiley & Sons, Inc.

1. THE OWA OPERATOR

Yager introduced the ordered weighted averaging (OWA) operator[1] to provide a parameterized family of mean type aggregators. Since its appearance in 1988 it has been used in an astonishingly wide variety of different applications.[1–16] Recently, Mitchell and Estrakh[17,18] and Schaefer and Mitchell[19] described a new generalized OWA‡ operator as follows.

Consider a set of input arguments a_i, $i \in \{1, 2, \dots, M\}$. We assume that each a_i has a priority α_i. We rearrange the inputs a_i according to the priorities α_i, such that the inputs with higher priority values come first. Suppose $a_{(k)}$ and $\alpha_{(k)}$§ denote the input arguments and priorities rearranged in order of the

*Author to whom correspondence should be addressed. e-mail: mitchell@is.elta.co.il
†e-mail: schaef@is.elta.co.il

‡The new OWA operator was first described by Mitchell and Estrakh[17] in 1997 where it is referred to as a modified OWA operator. The name induced OWA was first used by Yager and Filev.[20] This name, being more descriptive than modified OWA, will be used from now on. The induced operator itself was generalized by Mitchell and Estrakh[18] and Schaefer and Mitchell,[19] who allowed the priorities to be fuzzy numbers (see later on). In Ref. 19, the name generalized OWA was given to an induced OWA with fuzzy priorities.

§Generally we denote arguments in the input lists using the subscripts i or j, e.g., a_i or a_j. For the arguments in the rearranged lists we use the subscripts k or l enclosed in parentheses, e.g., $a_{(k)}$ or $a_{(l)}$.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 15, 317–327 (2000)

priorities α_i. If α_i is the kth largest priority, then

$$\begin{aligned} a_{(k)} &= a_i \\ \alpha_{(k)} &= \alpha_i \end{aligned} \tag{1}$$

The induced OWA operator is

$$F = \sum_{k=1}^{M} w_k a_{(k)} \tag{2}$$

where w_k, $k \in \{1, 2, \ldots, M\}$ are a set of weights such that

$$w_k \in [0, 1] \tag{3}$$

$$\sum_{k=1}^{M} w_k = 1 \tag{4}$$

Note: Suppose α_i and a_i are, respectively, the kth largest priority, and the lth largest input argument. Then generally $l \neq k$. If, however, for all i, $l = k$, then we obtain the original OWA operator of Yager.[1]

We originally proposed[17] the induced OWA operator within the context of predictive picture compression. Since then it has already been used in several other fields: granular computing,[20,21] fusion of expert information,[22] and noise smoothing.[19] Yager and Filev[20] have also investigated the mathematical properties of the induced OWA operator.

Note: the nomenclature used by Yager and Filev[20] differs from that used here and in Refs. 17–19: In Ref. 19 the a_i and α_i are referred to as the *input arguments* and *priorities*. In Ref. 20 the corresponding names are *argument variables* and *order inducing variables*.

The induced OWA operator may be written in compact form as a matrix equation: Let **w**, **a**, and $\boldsymbol{\alpha}$ be the column vectors

$$\begin{aligned} \mathbf{w} &= (w_1, w_2, \ldots, w_M)^T \\ \mathbf{a} &= (a_1, a_2, \ldots, a_M)^T \\ \boldsymbol{\alpha} &= (\alpha_1, \alpha_2, \ldots, \alpha_M)^T \end{aligned}$$

then Eq. 2 becomes

$$F = \mathbf{w}^T \mathbf{R} \mathbf{a} \tag{5}$$

where $\mathbf{R}$ is the *binary* permutation matrix whose elements R_{ki}, $i,k \in \{1,2,\ldots,M\}$, satisfy Eqs. 6–9:

$$R_{ki} = \begin{cases} 1 & \text{if } \alpha_i \text{ is the } k\text{th largest priority} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$R_{ki} \in \{0,1\} \tag{7}$$

$$\sum_{k=1}^{M} R_{ki} = 1 \tag{8}$$

$$\sum_{i=1}^{M} R_{ki} = 1 \tag{9}$$

EXAMPLE 1. Given the vectors

$$\mathbf{w} = (0.6\ \ 0.3\ \ 0.1)^T$$

$$\mathbf{a} = (-4\ \ 5\ \ 3)^T$$

$$\boldsymbol{\alpha} = (9\ \ 8\ \ 10)^T$$

the permutation matrix is

$$\mathbf{R} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

and the induced OWA aggregation is

$$F = \mathbf{w}^T\mathbf{Ra} = (0.6 \quad 0.3 \quad 0.1)\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} -4 \\ 5 \\ 3 \end{pmatrix} = 1.1$$

Mitchell and Estrakh[17] did not explicitly deal with ties, i.e., when two or more priorities have exactly the same value. Yager and Filev,[20] gave a simple prescription for handling such situations. Suppose two priorities, α_h and α_k, have exactly the same value. Then they define F by applying Eq. 2 to new arguments a'_i, where

$$a'_i = \begin{cases} a_i & \text{if } i \neq h,k \\ \frac{1}{2}(a_h + a_k) & \text{otherwise} \end{cases} \tag{10}$$

A more general approach for dealing with ties is to regard the priorities α_i as fuzzy numbers $\tilde{\alpha}_i$. In this case, the induced OWA operator becomes

$$F = \mathbf{w}^T\mathbf{Sa} \tag{11}$$

where **S** is a *fuzzy* permutation matrix whose elements, S_{ki}, $i, k \in \{1, 2, \ldots, M\}$, satisfy Eqs. 12–14:

$$S_{ki} \in [0, 1] \tag{12}$$

$$\sum_{k=1}^{M} S_{ki} = 1 \tag{13}$$

$$\sum_{l=1}^{M} S_{ki} = 1 \tag{14}$$

In Eq. 11 there is no need to define new arguments a'_i since the problem of ties is dealt with implicity: As the width of the fuzzy priorities tend to zero the Yager–Filev prescription (Eq. 10) is recovered in a natural way.

2. NEW OWA OPERATOR

The basic assumption which underlies Refs. 17–20 is that there is only *one* set of priorities, albeit they may be fuzzy numbers.[18,19] In this paper we discard this assumption by first assuming there are N different crisp priority vectors which we label $\boldsymbol{\alpha}_p$, $p \in \{1, 2, \ldots, N\}$. In this case the output in Eq. 2 is a vector **F**

$$\begin{aligned} \mathbf{F} &= (F_1 \;\; F_2 \;\; \cdots \;\; F_N) \\ &= \left(\mathbf{w}^T \mathbf{R}_1 \mathbf{a} \quad \mathbf{w}^T \mathbf{R}_2 \mathbf{a} \quad \cdots \quad \mathbf{w}^T \mathbf{R}_N \mathbf{a}\right) \end{aligned} \tag{15}$$

where $\mathbf{R}_p$, $p \in \{1, 2, \ldots, N\}$ is the permutation matrix corresponding to the priority vector $\boldsymbol{\alpha}_p$. We may collapse the vector **F** into a single scalar value ϕ by aggregating the outputs F_p. For this purpose we have many different choices of aggregation operators.

2.1. Weighted Average

In this case

$$\phi = \sum_{p=1}^{N} \Omega_p F_p \tag{16}$$

where Ω_p are a set of weights which satisfy Eqs. 12 and 13:

$$\Omega_p \in [0, 1] \tag{17}$$

$$\sum_{p=1}^{N} \Omega_p = 1 \tag{18}$$

Eq. 16 may be written in a more transparent form by substituting Eq. 5 into Eq. 16

$$\phi = \sum_{p=1}^{N} \Omega_p F_p = \mathbf{w}^T \sum_{p=1}^{N} \Omega_p \mathbf{R}_p \mathbf{a} = \mathbf{w}^T \mathbf{T} \mathbf{a} \tag{19}$$

where

$$\mathbf{T} = \sum_{p=1}^{N} \Omega_p \mathbf{R}_p \tag{20}$$

The matrix **T** is doubly stochastic in which the rows and columns sum to one. **T** is thus formally equivalent to **S**, although the considerations leading to **S** and **T** are quite different. Like **S**, the matrix **T** may be regarded as a fuzzy or probabilistic permutation matrix in which the (i, j)th element T_{ij} denotes the degree or probability that the argument a_j has the ith largest *average* priority. In the most general case, $N = M!$ since there are $M!$ permutation matrices $\mathbf{R}_i$. However it is not mandatory to use all $M!$ matrices and normally $N < M!$.

EXAMPLE 2. Let $M = 3$. Then there are six different permutation matrices

$$\mathbf{R}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{R}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{R}_3 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{R}_4 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad \mathbf{R}_5 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \mathbf{R}_6 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Suppose the weights Ω_i are

$$\Omega_1 = 0.1, \quad \Omega_2 = 0.15, \quad \Omega_3 = 0.4,$$
$$\Omega_4 = 0.05, \quad \Omega_5 = 0.1, \quad \Omega_6 = 0.2$$

then the matrix **T** is

$$\mathbf{T} = \sum_{p=1}^{N} \Omega_p \mathbf{R}_p = \begin{pmatrix} 0.25 & 0.45 & 0.30 \\ 0.60 & 0.20 & 0.20 \\ 0.15 & 0.35 & 0.50 \end{pmatrix}$$

2.2. OWA Operator

Suppose we rearrange the F_p values according to some criteria and let $F_{(q)}$ be the qth F_p value in the rearranged list. Then

$$\phi = \sum_{q=1}^{N} \Omega_q F_{(q)} \tag{21}$$

In this case

$$\mathbf{T} = \sum_{q=1}^{N} \Omega_q \mathbf{R}_{(q)} \tag{22}$$

where $\mathbf{R}_{(q)}$ is the permutation matrix corresponding to $F_{(q)}$. In this case $\mathbf{T}$ is still a stochastic matrix which we may continue to interpret as a fuzzy or probabilistic permutation matrix.

2.3. Multiple Fuzzy Priority Vectors

In Eqs. 20 and 22 we have assumed a set of crisp priority vectors $\boldsymbol{\alpha}_p$, $p \in \{1, 2, \ldots, N\}$. However, we may just as well assume fuzzy priority vectors $\tilde{\boldsymbol{\alpha}}_p$. In this case, Eqs. 20 and 22 remain unchanged except for the replacement of the binary matrices $\mathbf{R}_p$ by fuzzy matrices $\mathbf{S}_p$:

$$\mathbf{T} = \sum_{p=1}^{N} \Omega_p \mathbf{S}_p \tag{23}$$

$$\mathbf{T} = \sum_{q=1}^{N} \Omega_q \mathbf{S}_{(q)} \tag{24}$$

3. APPLICATIONS OF THE NEW OWA OPERATOR

We give two applications of the new OWA operator. The first application uses a weighted average of the F_i (Eqs. 19 and 20) and the second application uses an OWA of the $F_{(q)}$ (Eqs. 21 and 22).

3.1. Lossless Picture Compression

The induced OWA was first used in a predictive picture compression algorithm.[17] We now show how the new multiple priorities OWA operator may be used in a natural way to further improve the picture compression.

We start by briefly describing the method of differential pulse code modulation (DPCM) for lossless picture compression. At the transmitter, the rows in the input picture are scanned from top to bottom. In each row the pixels are scanned from left to right. At each pixel (m, n), the previously transmitted pixel gray-levels are used to predict the true gray-level. If $h(m, n)$ and $g(m, n)$ are, respectively, the predicted and true pixel gray-level, then the prediction residual $d(m, n) = h(m, n) - g(m, n)$ is encoded using a standard entropy encoder.

At the receiver the reverse process is followed for each pixel (m, n): the prediction $h(m, n)$ is calculated from the gray-levels of the previously transmitted pixels. The decoded difference $d(m, n)$ is then added to $h(m, n)$ to give the true gray-level $g(m, n)$.

In the median lossless picture compression algorithm, the prediction $h(m,n)$ for the present pixel (m,n) is calculated by using the median of three "basic" gray-level predictions $a_1(m,n)$, $a_2(m,n)$, and $a_3(m,n)$:

$$h(m,n) = \text{median}\,\{a_1(m,n), a_2(m,n), a_3(m,n)\} \tag{25}$$

where $a_1(m,n) = g(m-1,n)$, $a_2(m,n) = g(m,n-1)$, and $a_3(m,n) = g(m-1, n-1)$. First we note that the median algorithm may be reformulated using the original OWA operator of Yager[1],

$$h_{\mathrm{OWA}}(m,n) = \mathbf{w}_{\mathrm{OWA}}^T \mathbf{R}_{\mathrm{OWA}} \mathbf{a}$$

where $\mathbf{a} = (a_1\ a_2\ a_3^T)$, $\mathbf{w}_{\mathrm{OWA}} = (w_1\ w_2\ w_3^T) = (0\ 1\ 0^T)$, and $\mathbf{R}_{\mathrm{OWA}}$ is the permutation matrix which rearranges the a_i in order of decreasing value. *Note*: we append the subscript OWA to h, $\mathbf{w}$, and $\mathbf{R}$ to emphasize that these quantities are calculated according to the original OWA operator of Yager.[1]

To improve the compression performance, we follow Mitchell and Estrakh[17] and rearrange the three basic predictions $a_1(m,n), a_2(m,n), a_3(m,n)$ according to how well the corresponding previous pixel predictions $a_1(m-1,n)$, $a_2(m-1,n)$, $a_3(m-1,n)$ performed for the previous pixel $(m-1,n)$. We do this by setting α_i equal to the negative absolute error between the ith basic predictor for the previous pixel and the gray-level of the previous pixel:

$$\alpha_i = -|a_i(m-1,n) - g(m-1,n)|$$

We then form a new prediction $h_{\mathrm{IND}}(m,n)$ for the present pixel (m,n) using the induced OWA operator,

$$h_{\mathrm{IND}}(m,n) = \mathbf{w}_{\mathrm{IND}}{}^T \mathbf{R}_{\mathrm{IND}} \mathbf{a}$$

where $\mathbf{R}_{\mathrm{IND}}$ is the permutation matrix which rearranges the a_i according to their priorities α_i.

We choose the weight vector $\mathbf{w}_{\mathrm{IND}}$ for minimum zero-order entropy E,

$$E = -\sum_h p_h \log(p_h)$$

where p_h is the probability of occurrence of a gray-level difference $d(m,n) = h(m,n) - g(m,n)$. In practice we found that using the previous best predictor,|| i.e., using the weight vector

$$\mathbf{w}_{\mathrm{IND}} = (1\ 0\ 0)^T$$

gave very good results.

Mitchell and Estrakh[17] used only one priority vector $\boldsymbol{\alpha}$ which measured the performance of the basic predictors at the previous pixel $(m-1,n)$. However, there are in fact many possible choices of priority vectors. The only constraint is that the performance of the basic predictor is measured at a pixel which is close

||Yager and Filev[20] refer to this as the best yesterday model.

to (m, n) and which has already been transmitted. Thus we may use the four pixels $(m, n-1)$, $(m-1, n-1)$, $(m-1, n)$, and $(m-1, n+1)$ to define four different priority vectors $\boldsymbol{\alpha}_1 = (\alpha_{11}\ \alpha_{12}\ \alpha_{13})$, $\boldsymbol{\alpha}_2 = (\alpha_{21}\ \alpha_{22}\ \alpha_{23})$, $\boldsymbol{\alpha}_3 = (\alpha_{31}\ \alpha_{32}\ \alpha_{33})$, and $\boldsymbol{\alpha}_4 = (\alpha_{41}\ \alpha_{42}\ \alpha_{43})$, where

$$\alpha_{1i} = -|a_i(m, n-1) - g(m, n-1)|$$

$$\alpha_{2i} = -|a_i(m-1, n-1) - g(m-1, n-1)|$$

$$\alpha_{3i} = -|a_i(m-1, n) - g(m-1, n)|$$

$$\alpha_{4i} = -|a_i(m-1, n+1) - g(m-1, n+1)|.$$

The predicted gray-level at (m, n) is thus

$$h_{\mathrm{NEW}}(m, n) = \mathbf{w}_{\mathrm{NEW}}^T \left(\sum_{p=1}^{4} \Omega_p \mathbf{R}_{\mathrm{NEW},\,p} \right) \mathbf{a}$$

In the experiments reported in this paper we set all four weights Ω_p equal to 0.25 and the final equation for the predicted gray-level is

$$h_{\mathrm{NEW}}(m, n) = \frac{\mathbf{w}_{\mathrm{NEW}}^T}{4} \left(\sum_{p=1}^{4} \mathbf{R}_{\mathrm{NEW},\,p} \right) \mathbf{a}$$

We compare the compression efficiency of the new predictor with the original and induced OWA predictors by encoding six test pictures and measuring the corresponding entropies E_{NEW}, E_{OWA}, and E_{IND}. In these experiments we used the weight vectors:

$$\mathbf{w}_{\mathrm{NEW}} = (1\ 0\ 0)^T$$

$$\mathbf{w}_{\mathrm{OWA}} = (0\ 1\ 0)^T$$

$$\mathbf{w}_{\mathrm{IND}} = (1\ 0\ 0)^T$$

Table I gives the zero-order entropies obtained with the different predictors. We see that for all six test pictures, the new algorithm always had the

Table I. Zero-order entropies for original, induced and new OWA predictors.

Picture Name	Original	Induced	New
Board	3.16	2.80	2.71
Balloon	2.50	2.28	2.15
Barb2	4.03	3.75	3.62
Boats	3.46	3.13	3.03
Gold	3.621	3.43	3.31
Hotel	3.69	3.36	3.25

lowest zero-order entropy. In particular we note the consistent reduction in entropy when we use a multiple priority induced OWA instead of a single priority induced OWA operator.

3.2. Multiple-Attribute Classifier

The K-nearest neighbor (K-NN) classifier is a voting scheme which is widely used in problems involving pattern recognition and classification. A typical problem requires finding F: the degree to which an unknown pattern X belongs to a given class C.

In the K-NN algorithm we suppose that a small subset of M patterns, $Q_1, Q_2, \ldots, Q_M$, have been classified beforehand by a panel of experts. Each pattern Q_i and the unknown pattern X are characterized by N different attributes f_p, $p \in \{1, 2, \ldots, N\}$. Let $\alpha_{pi} = -|f_{pi} - f_{px}|$ where γ_{pi} and γ_{px} are, respectively, the pth attribute value of Q_i and the pth attribute value of X. Then it is common practice to combine the differences α_{pi}, $p \in \{1, 2, \ldots, N\}$ into a single priority value b_i by using the negative of the Mahalanobis or of the Euclidean distance:

$$b_i = -\sqrt{\alpha_{1i}{}^2 + \alpha_{2i}{}^2 + \cdots + \alpha_{Ni}{}^2} \tag{26}$$

Given the unknown pattern X we rearrange the Q_i in according to their priorities b_i. If $Q_{(k)}$ is the known pattern with the kth largest priority, i.e.,

$$Q_{(k)} = Q_i \qquad \text{if} \qquad b_i \text{ is the } k\text{th largest priority}$$

then F is defined as

$$F = \frac{1}{K} \sum_{k=1}^{K} a_{(k)}$$

where K is a parameter defined by the user and $a_{(k)}$ is the degree to which, according to the panel of experts, the pattern $Q_{(k)}$ belongs to class C.

Mathematically we may write the algorithm in terms of the induced OWA operator

$$F = \mathbf{w}^T \mathbf{R} \mathbf{a} \tag{27}$$

where

$$\mathbf{a} = (a_1, a_2, \ldots, a_M)^T$$

$$\mathbf{w} = \frac{1}{K} \Big(\overbrace{11 \cdots 1}^{K} \overbrace{00 \cdots 0}^{M-K} \Big)^T$$

and $\mathbf{R}$ is the $M \times M$ binary permutation matrix which rearranges the elements a_i according to their priorities b_i.

A major problem with the K-NN voting scheme is that the different attributes of the object are not usually commensurate one with the other. This in turn makes the process of combining the N attributes into a single priority vector both difficult and artificial.

We now describe an alternative approach which does not rely on combining the N attributes into a single priority vector. The new approach is based on the new multiple priority induced OWA and works as follows.

For each attribute $p \in \{1, 2, \ldots, N\}$ we define a corresponding priority vector $\boldsymbol{\alpha}_p$, where

$$\boldsymbol{\alpha}_p = (\alpha_{p1} \;\; \alpha_{p2} \;\; \cdots \;\; \alpha_{pM})^T$$

For each attribute p we use the single priority induced OWA operator (Eq. 27) to calculate F_p:

$$F_p = \mathbf{w}^{\mathrm{T}} \mathbf{R}_p \mathbf{a}$$

The final classification of P, i.e., the overall degree to which P belongs to the class C, is ϕ, which is found by aggregating the individual F_p. Very often we would use a pessimistic aggregation, i.e.,

$$\begin{aligned}\phi &= \min(F_1, F_2, \ldots, F_N) \\ &= \sum_{q=1}^{N} \Omega_q F_{(q)}\end{aligned}$$

where $F_{(q)}$ is the qth largest F_i value and $\boldsymbol{\Omega} = (000 \cdots 01)^T$. More generally we may use a less pessimistic aggregation by appropriately choosing the weight vector $\boldsymbol{\Omega}$.

4. CONCLUSION

In this paper, we described an induced OWA operator with multiple priorities. We described two applications of the new OWA operator: predictive picture compression and multiple-attribute classification. The authors plan to give several other applications of the multiple priority induced OWA operator in the near future.

References

1. Yager RR. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. IEEE Trans Syst Man Cybernet 1988;SMC-18:183–190.
2. Yager RR. Applications and extensions of OWA aggregation. Int J Man-Machine Stud 1992;37:103–132.
3. Cutello V, Montero J. Hierarchies of aggregation operators. Int J Intell Syst 1994;9:1025–1045.
4. Cho S-B. Fuzzy aggregation of modular neural networks with ordered weighted averaging operators. Int J Approx Reasoning 1995;13:359–375.

5. Yager RR. On a semantics for neural networks based on fuzzy quantifiers. Int J Intell Syst 1992;7:765–786.
6. Bordogna G, Pasi G. Linguistic aggregation operators of selection criteria in fuzzy information retrieval. Int J Intell Syst 1995;10:233–248.
7. Yager R.R. Fuzzy quotient operators for fuzzy relational data bases. In: Proceedings of the International Fuzzy Engineering Symposium, Yokohama, 1991. p 281–296.
8. Yager RR. A note on weighted queries in information retrieval systems. J Am Soc Inform Sci 1987;28:23–24.
9. Schaefer PA, Mitchell HB, Estrakh DD. A fuzzy kth ordered statistic. Int J Intell Syst 1999;14:211–221.
10. Yager RR, Filev DP, Sadeghi T. Analysis of flexible structured fuzzy logic controllers. IEEE Trans Syst Man Cybernet 1994;24:1035–1043.
11. O'Hagan M. Aggregating template or rule antecedents in real-time expert systems with fuzzy set logic. In: Proceedings of the 22nd Annual IEEE Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, 1988. p 681–689.
12. Kacprzyk J. Inductive learning from considerably erroneous examples with a specificity based rule. In: Proceedings of the International Conference on Fuzzy Logic and Neural Networks, Iizuka, Japan, 1990. p 819.
13. Yager RR, Goldstein LS, Mendels E. FUZMAR: an approach to aggregating market research data based on fuzzy reasoning. Fuzzy Sets Syst 1994;68:1–11.
14. Yager RR. Quantifier guided aggregation using OWA operators. Int J Intell Syst 1996;11:49–73.
15. Yager RR. Interpreting linguistically quantified propositions. Int J Intell Syst 1994;9:541–569.
16. Yager RR. Solving mathematical programming problems with OWA operators as objective functions. In: Proceedings of the 1995 IEEE International Conference Fuzzy Systems, pp. 1441–1446.
17. Mitchell HB, Estrakh DD A modified OWA operator and its use in lossless DPCM image compression. J Uncertainty, Fuzziness Knowledged-based Syst 1997;5:429–436.
18. Mitchell HB, Estrakh DD. An OWA operator with fuzzy ranks. Int J Intell Syst 1998;13:69–81.
19. Schaefer PA, Mitchell HB. A generalized OWA operator. Int J Intell Syst 1999;14:123–144.
20. Yager RR, Filev DP. Induced OWA operators. IEEE Syst Cybernet Part B, 1999;29:141–150.
21. Yager RR, Filev DP. Operations for granular computing: mixing words and numbers. In: 1998 IEEE International Conference on Fuzzy Systems Proceedings, 1998;1:123–128.
22. Yager RR, Lan R. A new approach to the fusion of expert information. In: Proceedings of the IEEE/IAFE/INFORMS 1998 Conference on Computational Intelligence for Financial Engineering, 1998. p 220–223.