Fuzzy Sets and Systems 94 (1998) 157–169

On the issue of obtaining OWA operator weights

Dimitar Filev[a], Ronald R. Yager[b,*]

[a] *Ford Motor Company, Detroit, MI 48239, USA*
[b] *Machine Intelligence Institute, Iona College, New Rochelle, NY 10801, USA*

Received January 1995; revised August 1996

Abstract

We first investigate the issue of obtaining the weights associated with the OWA aggregation in the situation when we have observed data on the arguments and the aggregated value. We next introduce a family of OWA operators called exponential OWA operators. Finally, we look at a simple procedure for generating the weights given a required degree of orness. © 1998 Elsevier Science B.V.

1. Introduction

The concept of ordered weighted operators (OWA) was introduced by Yager [7]. A class of OWA operators called S-OWA operators was introduced in [12]. Some new families of OWA operators were discussed by Yager [10]. Several applications of the OWA operators were reported during the short time period following their first appearance: in decision making [3, 9], expert systems [4], neural networks [2, 8], fuzzy systems and control [11], and communication networks [6].

One important issue in the theory of OWA aggregation is the determination of the associated weights. A number approaches have been suggested for obtaining the weights [10]. One of the first methodologies for obtaining the weights was developed by O'Hagan [5]. O'Hagan's approach allows one to calculate the vector of the OWA weights for a predefined level of *orness* (optimism); among the variety of possible solutions to this problem O'Hagan selects the vector which maximizes the entropy of the OWA weights; algorithmically, it is based on the solution of a constrained optimization problem. Thus the method of O'Hagan in fact determines a special class of OWA operators having maximal entropy of the OWA weights for a given level of *orness*.

In this paper we focus on some issues related to the acquisition of the OWA weights. We first develop an algorithm for the calculation of the OWA weights that allows us to learn the weights from data consisting of tuples of individual scores along with their aggregated value. Next we introduce also a class of OWA

* Corresponding author. Tel.: +1 212 249 2047; fax: +1 212 249 1689; e-mail: ryager@iona.edu.

0165-0114/98/$19.00 © 1998 Elsevier Science B.V. All rights reserved
PII S0165-0114(96)00254-0

operators, called *Exponential OWA Operators*, whose weights can be calculated in a very simple way for a given level of *orness*. Further we demonstrate an effective mechanism for generation of OWA operators.

2. The concept of ordered weighted averaging operators

An OWA operator [1] of dimension n is a mapping:

$$f: R^n \to R,$$

that has an associated weighting vector W

$$W = [w_1 \ w_2 \ \dots \ w_n]^{\mathrm{T}}$$

such that

$$\sum_i w_i = 1; \quad w_i \in [0, 1]$$

and where

$$f(a_1, \dots, a_n) = \sum_{j=1}^{n} w_j b_j,$$

where b_j is the jth largest element of the collection of the aggregated objects $a_1, a_2, \dots, a_n$. The function value $f(a_1, \dots, a_n)$ determines the aggregated value of arguments, $a_1, a_2, \dots, a_n$.

A fundamental aspect of the OWA operator is the re-ordering step, in particular an argument a_i is *not* associated with a particular weight w_i but rather a weight w_i is associated with a particular ordered position i of the arguments. A known property of the OWA operators is that they include the Max, Min and arithmetic mean operators for the appropriate selection of the vector W:

1. For $W = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, $f(a_1, \dots, a_n) = \operatorname{Max}_i a_i$

2. For $W = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$, $f(a_1, \dots, a_n) = \operatorname{Min}_i a_i$

3. For $W = \begin{bmatrix} 1/n \\ 1/n \\ \vdots \\ 1/n \end{bmatrix}$, $f(a_1, \dots, a_n) = \dfrac{1}{n} \sum_{i=1}^{n} a_i.$

It can be easily shown [1] that the OWA operators are aggregation operators, satisfying the commutativity, monotonicity and idempotency properties and that they are bounded by the Max and Min operators, for

OWA operators

$$\operatorname*{Min}_i a_i \leqslant f(a_1, \dots, a_n) \leqslant \operatorname*{Max}_i a_i.$$

Since this class of operators runs between the Max (or) and the Min (and) in [7] Yager introduced a measure to characterize the type of aggregation being performed for a particular value of the weighting vector. This measure called the *orness measure* of the aggregation is defined as

$$\text{orness}(W) = \frac{1}{n-1}\sum_{i=1}^{n}(n-i)w_i. \tag{1}$$

As suggested by Yager [7] this measure, which lies in the unit interval, characterizes the degree to which the aggregation is like an *or* (Max) operation. It can be shown that

$$\text{orness}([1\ \ 0\ \ \dots\ \ 0]^{\mathrm{T}}) = 1,$$

$$\text{orness}([0\ \ 0\ \ \dots\ \ 1]^{\mathrm{T}}) = 0,$$

$$\text{orness}([1/n\ \ 1/n\ \ \dots\ \ 1/n]^{\mathrm{T}}) = 0.5.$$

Therefore the Max, Min and arithmetic mean operators can be regarded as OWA operators with degree of orness, respectively, 1, 0 and 0.5.

A second measure introduced by Yager [7] was the dispersion or entropy associated with a weighting vector

$$\text{Disp}(W) = \sum_{i=1}^{n} w_i \ln w_i.$$

This was suggested for use in calculating how much of the information in the arguments is used during an aggregation based on W.

In [5] O'Hagan used these measures to develop a procedure to generate the OWA weights that have a predefined degree of orness α and maximize the entropy. O'Hagan called them MEOWA operators. The approach suggested by O'Hagan is based on the solution of the following constrained optimization problem [5]:

$$\text{Maximize} \sum_{i=1}^{n} w_i \ln w_i$$

$$\text{subject to } \alpha = \frac{1}{n-1}\sum_{i=1}^{n}(n-i)w_i,$$

$$\sum_{i=1}^{n} w_i = 1,$$

$$w_i \in [0, 1], \quad i = (1, \dots, n).$$

We note that for this we need to have specified the desired degree of orness α.

3. Learning OWA operators from observations

In this section we shall suggest an algorithm which can be used to learn the weights associated with a particular use of the OWA operator from an observation of the performance of some agent. Formally, we shall assume the following information is available. Given are a collection of m samples (observations) each comprised of an n-tuple of values $(a_{k1}, a_{k2}, \dots, a_{kn})$, called the arguments, and an associated single value called the aggregated value, which we shall denote as d_k. A prototypical situation which can generate such a data set will be described as follows. Assume we have a set of m alternatives and a set of n criteria which are used to evaluate each of the alternatives. We indicate these scores as a_{ij}. An expert reviews the scores obtained by each alternative on the n criteria and then provides an aggregate score for that alternative denoted d_k.

Our goal will be to obtain an OWA operator, a weighting vector W that models the process of aggregation used in that data set. We need a OWA operator, W, such that for the entire collection of data we as faithfully as possible satisfy the condition

$$f(a_{k1}, a_{k2}, \dots, a_{kn}) = d_k$$

for any k.

This problem can be simplified by taking advantage of the linearity of the OWA aggregation with respect to the ordered arguments. We denote the reordered objects of the kth sample by $b_{k1}, b_{k2}, \dots, b_{kn}$ where b_{kj} is the jth largest element of the argument collection $a_{k1}, a_{k2}, \dots, a_{kn}$. Using these ordered arguments the problem of modelling the aggregation process is to find the vector of OWA weights $W = [w_1 \ w_2 \ \dots \ w_n]^T$ such that

$$b_{k1}w_1 + b_{k2}w_2 + \cdots + b_{kn}w_n = d_k$$

for any $k = 1, \dots, m$.

We shall relax this formulation by looking for a vector of OWA weights $W = [w_1 \ w_2 \ \dots \ w_n]^T$ that approximates the aggregation operator by minimizing the instantaneous errors e_k where

$$e_k = \tfrac{1}{2}(b_{k1}w_1 + b_{k2}w_2 + \cdots + b_{kn}w_n - d_k)^2$$

with respect to weights w_i. The solution of this problem seems to be simple and one could expect that it can be done by application of the Widrow–Hoff rule [13]. The situation is complicated by the fact that the above minimization problem is a constrained optimization problem, since the OWA weights w_i have to satisfy the following two properties:

1. $\sum_{i=1}^{n} w_i = 1;$
2. $w_i \in [0, 1], \qquad i = (1, \dots, n).$

To avoid the constraints on w_i we assume that the OWA weights are defined as follows:

$$w_i = \frac{e^{\lambda_i}}{\sum_{j=1}^{n} e^{\lambda_j}}, \qquad i = (1, \dots, n).$$

From the above transformation it becomes clear that for any values of the parameters λ_i the weights w_i will be positive and will sum to 1. Therefore the constrained minimization problem is transformed to the following unconstrained nonlinear programming problem:

Minimize the instantaneous errors e_k:

$$e_k = \frac{1}{2}\left(b_{k1}\frac{e^{\lambda_1}}{\sum_{j=1}^{n} e^{\lambda_j}} + b_{k2}\frac{e^{\lambda_2}}{\sum_{j=1}^{n} e^{\lambda_j}} + \cdots + b_{kn}\frac{e^{\lambda_n}}{\sum_{j=1}^{n} e^{\lambda_j}} - d_k\right)^2$$

with respect to the parameters λ_i.

We shall use the gradient descent technique to solve this problem. Using the gradient descent method we obtain the following rule for updating the parameters λ_i, $i = (1, \dots, n)$:

$$\lambda_i(l+1) = \lambda_i(l) - \beta \left.\frac{\partial e_k}{\partial \lambda_i}\right|_{\lambda_i = \lambda_i(l)},$$

where β denotes the learning rate ($0 \leqslant \beta \leqslant 1$).

For notational simplification we shall denote by $\hat{d}_k$ the estimate of the aggregated value d_k:

$$\hat{d}_k = b_{k1}\frac{e^{\lambda_1}}{\sum_{j=1}^n e^{\lambda_j}} + b_{k2}\frac{e^{\lambda_2}}{\sum_{j=1}^n e^{\lambda_j}} + \cdots + b_{kn}\frac{e^{\lambda_n}}{\sum_{j=1}^n e^{\lambda_j}}.$$

Then for the partial derivative $\partial e_k/\partial \lambda_1$ we get

$$\frac{\partial e_k}{\partial \lambda_1} = \left(b_{k1}\frac{e^{\lambda_1}\sum_{j=1}^n e^{\lambda_j} - e^{2\lambda_1}}{(\sum_{j=1}^n e^{\lambda_j})^2} + b_{k2}\frac{-e^{\lambda_1}e^{\lambda_2}}{(\sum_{j=1}^n e^{\lambda_j})^2} + \cdots + b_{kn}\frac{-e^{\lambda_1}e^{\lambda_n}}{(\sum_{j=1}^n e^{\lambda_j})^2}\right)(\hat{d}_k - d_k),$$

$$\frac{\partial e_k}{\partial \lambda_1} = \left(b_{k1}\frac{e^{\lambda_1}\sum_{j=2}^n e^{\lambda_j}}{(\sum_{j=1}^n e^{\lambda_j})^2} + b_{k2}\frac{-e^{\lambda_1}e^{\lambda_2}}{(\sum_{j=1}^n e^{\lambda_j})^2} + \cdots + b_{kn}\frac{-e^{\lambda_1}e^{\lambda_n}}{(\sum_{j=1}^n e^{\lambda_j})^2}\right)(\hat{d}_k - d_k)$$

$$\frac{\partial e_k}{\partial \lambda_1} = \frac{e^{\lambda_1}}{\sum_{j=1}^n e^{\lambda_j}}\left[\frac{e^{\lambda_2}}{\sum_{j=1}^n e^{\lambda_j}}(b_{k1} - b_{k2}) + \cdots + \frac{e^{\lambda_n}}{\sum_{j=1}^n e^{\lambda_j}}(b_{k1} - b_{kn})\right](\hat{d}_k - d_k).$$

This expression can be rewritten by using the substitution $w_i = e^{\lambda_i}/\sum_{j=1}^n e^{\lambda_j}$ into the form

$$\frac{\partial e_k}{\partial \lambda_1} = w_1\,[w_2(b_{k1} - b_{k2}) + \cdots + w_n(b_{k1} - b_{kn})](\hat{d}_k - d_k),$$

$$\frac{\partial e_k}{\partial \lambda_1} = w_1[(w_2 + \cdots + w_n)b_{k1} - (w_2 b_{k2} + \cdots + w_n b_{kn})](\hat{d}_k - d_k),$$

$$\frac{\partial e_k}{\partial \lambda_1} = w_1[(1 - w_1)b_{k1} - (w_2 b_{k2} + \cdots + w_n b_{kn})](\hat{d}_k - d_k),$$

$$\frac{\partial e_k}{\partial \lambda_1} = w_1[b_{k1} - (w_1 b_{k1} + w_2 b_{k2} + \cdots + w_n b_{kn})](\hat{d}_k - d_k),$$

$$\frac{\partial e_k}{\partial \lambda_1} = w_1(b_{k1} - \hat{d}_k)(\hat{d}_k - d_k).$$

In a similar manner we obtain for the other partial derivatives:

$$\frac{\partial e_k}{\partial \lambda_i} = w_i(b_{ki} - \hat{d}_k)(\hat{d}_k - d_k), \quad i = (1, n).$$

Then we derive the final form of the rule for updating the parameters λ_i; we get

$$\lambda_i(l+1) = \lambda_i(l) - \beta w_i(b_{ki} - \hat{d}_k)(\hat{d}_k - d_k), \tag{2}$$

where parameters w_i are calculated at each iteration step for the current values of parameters $\lambda_i(l)$:

$$w_i = \frac{e^{\lambda_i(l)}}{\sum_{j=1}^n e^{\lambda_j(l)}}, \quad i = (1, n) \tag{3}$$

and $\hat{d}_k$ is the current estimate of the aggregated values d_k:

$$\hat{d}_k = b_{k1}w_1 + b_{k2}w_2 + \cdots + b_{kn}w_n. \tag{4}$$

Therefore the parameters λ_i determining the OWA weights are updated by propagation of the error $(\hat{d}_k - d_k)$ between the current estimated aggregated value and the actual aggregated value with factors w_i and $(b_{ki} - \hat{d}_k)$. These factors are the current OWA weight w_i and the difference $(b_{ki} - \hat{d}_k)$ between the ith aggregate object b_{ki} and the current estimated aggregated value $\hat{d}_k$.

We shall illustrate the algorithm for learning OWA weights w_i (Eqs. (2–4)) in the following example.

Example 1. We assume the collection of samples of data in Table 1. Each sample consists of 4 argument values and the relevant aggregated value:

Table 1

Sample	Argument values				Aggregated value
1	0.4	0.1	0.3	0.8	0.24
2	0.1	0.7	0.4	0.1	0.16
3	1.0	0.0	0.3	0.5	0.15
4	0.2	0.2	0.1	0.4	0.17
5	0.6	0.3	0.2	0.1	0.18

The aggregated values for each sample were calculated by using the Hurwicz [3] method for compromise aggregation. According to this method, which is widely applied in decision making, the aggregated value d obtained from a tuple of n arguments, $a_1, a_2, \ldots, a_n$, is defined as a weighted average of the Max and Min values of that tuple

$$\rho \operatorname*{Max}_i a_i + (1-\rho)\operatorname*{Min}_i a_i = d,$$

where parameter ρ represents the optimism of the decision maker, $0 \leqslant \rho \leqslant 1$. For example the first data set was calculated using $\rho = 0.2$. From the Max and Min values for the first example we obtained

$$0.2(0.8) + (1-0.2)(0.1) = 0.24.$$

Slightly different values of parameter ρ were used for each argument tuple to reflect the reasonable variation that is possible with slightly different mechanisms of aggregation in different samples, which are due to the individuality of different experts. For the calculation of the aggregated values we used the following values of the parameter ρ:

$$\rho = 0.2;\ 0.1;\ 0.15;\ 0.25;\ 0.18.$$

The learning algorithm, Eqs. (2)–(4), was applied on the reordered argument tuples. The learning algorithm was started with initial values $\lambda_i(0) = 0$, $i = (1, 4)$, with initial values of the OWA weights $w_i = 0.25$. A learning rate of $\beta = 0.35$ was used. The estimated values of λ_i after 150 iterations were:

$$\lambda_1 = -0.61;\quad \lambda_2 = -0.29;\quad \lambda_3 = -0.05;\quad \lambda_4 = 1.51.$$

These values of the λ_i induce the following OWA weights:

$$w_1 = 0.08;\quad w_2 = 0.11;\quad w_3 = 0.14;\quad w_4 = 0.67.$$

Estimated aggregated values $\hat{d}_k$ at the and of the learning process were

$$\hat{d}_1 = 0.22; \quad \hat{d}_2 = 0.18; \quad \hat{d}_3 = 0.18; \quad \hat{d}_4 = 0.15; \quad \hat{d}_5 = 0.18.$$

Using the OWA weights learned in the above and applying formula (1) we calculated a degree of orness 0.199. We found it to be a reasonable characteristic of the total level of orness that is associated with the entire collection of samples, compared with the different levels of optimism for the individual samples.

4. On an exponential class of OWA operators

In this section we introduce a class of OWA operators which we shall call *exponential* OWA operators. We study their usefulness in the problem of generation of OWA weights satisfying a given degree of orness. One possible way of solving this problem was suggested by O'Hagan, but as we mentioned above it involves the solution of a constrained nonlinear programming problem. We shall see that for the exponential OWA operators a very simple relationship exists between the orness degree and the parameter which determines the OWA weights.

We shall consider a family of OWA operators whose weights are of the same form as the well-known set of weights that are used in the method of exponential smoothing [1]. We define the OWA weights as follows:

$$w_1 = \alpha; \quad w_2 = \alpha(1-\alpha); \quad w_3 = \alpha(1-\alpha)^2; \quad \ldots; \quad w_{n-1} = \alpha(1-\alpha)^{n-2}; \quad w_n = (1-\alpha)^{n-1}, \tag{5}$$

where parameter α belongs to the unit interval, $0 \leqslant \alpha \leqslant 1$. We can express these weights recursively as

1. $w_1 = \alpha \in [0, 1]$,
2. $w_j = w_{j-1}(1 - w_1), j = 2, \ldots, n-1$,
3. $w_n = w_{n-1}(1 - w_1)/w_1$.

There exists an alternative view to the process of generation of these weights. Assume W is a weighting vector of dimension n we extend it to a vector V of dimension $n+1$ as follows:

$$v_i = w_i \quad \text{for } i = 1, 2, \ldots, n-1,$$

$$v_n = \alpha w_n = w_1 w_n,$$

$$v_{n+1} = (1-\alpha)w_n = (1 - w_1)w_n.$$

Thus in extending this we simply proportionally divide the weight in the last place of the old vector.

It is evident that for the above defined weights it is true that $0 \leqslant w_i \leqslant 1$. In addition, the above weights satisfy the following identities:

$$w_{n-1} + w_n = (1-\alpha)^{n-2}(\alpha + 1 - \alpha) = (1-\alpha)^{n-2},$$

$$w_{n-2} + w_{n-1} + w_n = (1-\alpha)^{n-3}(\alpha + 1 - \alpha) = (1-\alpha)^{n-3},$$

$$\ldots$$

$$w_1 + w_2 + \cdots + w_n = 1.$$

Thus from a formal point of view these weights can serve as OWA weights.

For $\alpha = 1$ we obtain the following vector of OWA weights:

$$W = [1 \;\; 0 \;\; \ldots \;\; 0]^{\mathrm{T}}$$

and thus degree of orness 1. For $\alpha = 0$ we get the following vector of OWA weights:

$$W = [0 \;\; 0 \;\; \ldots \;\; 1]^{\mathrm{T}}$$

and therefore degree of orness 0. The orness of this OWA operator for different values of parameter α is determined by the formula (1):

$$\operatorname{orness}(W) = \frac{1}{n-1}\sum_{i=1}^{n}(n-i)w_i.$$

We note since $n - i = 0$ for $i = n$ then

$$\operatorname{orness}(W_n) = \frac{1}{n-1}\sum_{i=1}^{n}(n-i)w_i = \frac{1}{n-1}\sum_{i=1}^{n-1}(n-i)w_i.$$

Consider the orness value for $n+1$, $\operatorname{orness}(W_{n+1})$:

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

$$\operatorname{orness}(W_{n+1}) = \frac{n-1}{n}\operatorname{orness}(W_n) + \frac{1}{n}(1-w_n) + \frac{1}{n}\alpha w_n = \frac{n-1}{n}\operatorname{orness}(W_n) + \frac{1}{n}(1-w_{n+1}),$$

$$\operatorname{orness}(W_{n+1}) = \operatorname{orness}(W_n) + \frac{1}{n}[1 - \{\operatorname{orness}(W_n) + w_{n+1}\}].$$

We note that

$$\operatorname{orness}(W_n) = \frac{1}{(n-1)}\sum_{i=1}^{n}(n-i)w_i,$$

$$\operatorname{orness}(W_{n+1}) = \frac{1}{n}\sum_{i=1}^{n+1}(n+1-i)w_i = \frac{1}{n}\sum_{i=1}^{n}(n+1-i)w_i.$$

Consider

$$\frac{n+1-i}{n} - \frac{n-i}{n-1} = \frac{i-1}{n(n-1)} \geqslant 0$$

thus $\operatorname{orness}(W_{n+1}) \geqslant \operatorname{orness}(W_n)$, hence the orness value increases as n increases for a fixed n. It can also be proved that the orness function is a monotonically increasing function of parameter α.

The functional relationship between the orness and parameter α for different number of arguments $n = 2, 3, \ldots, 10$ is presented in Fig. 1. We note the monotonically increasing nature of this relationship with respect to α; for a fixed n the orness of this aggregation increases as α increases. We also note the monotonically increasing nature of this relationship with respect to n, for a fixed α the orness of this aggregation increases as n increases. Furthermore, it can be shown that for $n = 2$ the orness value of this aggregation is always equal to α. This fact along with the monotonicity with respect to n results in the notable feature that for any value of $n > 2$ the degree of orness is higher than the value of the parameter α. For this reason we shall call the OWA operators associated with these weights *optimistic exponential OWA operators*. In particular, we note that as the number of arguments increases this aggregation becomes more and more orlike.

An alternative related OWA operator can be derived by considering the following OWA weights:

$$w_1 = \alpha^{n-1};\quad w_2 = (1-\alpha)\alpha^{n-2};\quad w_3 = (1-\alpha)\alpha^{n-3};\quad \ldots;\quad w_{n-1} = (1-\alpha)\alpha;\quad w_n = (1-\alpha). \tag{6}$$

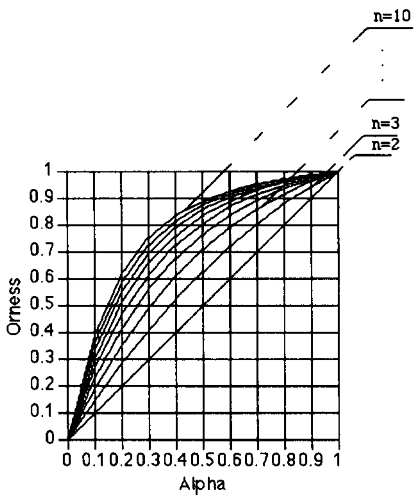

Fig. 1. Functional relationship between the orness of the optimistic exponential OWA operator and its parameter α for $n = 2, 3, \ldots, 10$.

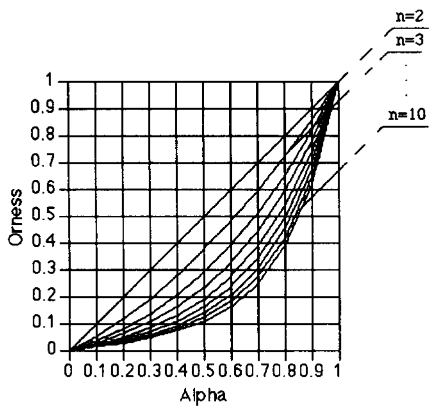

Fig. 2. Functional relationship between the orness of the pessimistic exponential OWA operator and its parameter α for $n = 2, 3, \ldots, 10$.

We can express these weights recursively as

1. $w_n = (1 - \alpha)$;
2. $w_j = w_{j+1}\alpha = w_j(1 - w_n)$, $j = 2, \ldots, n - 1$;
3. $w_1 = w_2(1 - w_n)/w_n$.

There exists an alternative view to the process of generation of these weights. Assume W is a weighting vector of dimension n we extend it to a vector V of dimension $n + 1$ as follows:

$$v_{i+1} = w_i \quad \text{for } i = 2, \ldots, n,$$

$$v_2 = (1 - \alpha)w_1 = w_n w_1,$$

$$v_1 = \alpha w_1 = (1 - w_n)w_1.$$

Thus in extending this we push down all the elements and simply proportionally divide the weight in the top place of the old vector to the two top places in the new vector.

In a similar manner as that for the optimistic exponential OWA operator, we can show that these weights satisfy the formal requirements to be OWA weights:

$$0 \leqslant w_i \leqslant 1, \quad i = (1, n) \text{ and } \sum_{i=1}^{n} w_i = 1.$$

We also note that for $\alpha = 0$ we obtain the pure Min and for $\alpha = 1$ we obtain the pure Max.

The functional relationship between the orness and parameter α for different numbers of aggregate objects $n = 2, 3, \ldots, 10$ is presented in Fig. 2. As in the previous case we note the monotonically increasing nature of this relationship with respect to α, for a fixed n the orness of this aggregation increases as α increases. Contrary to the previous case we also note the monotonically decreasing nature of this relationship with respect to n. For a fixed α the orness of this aggregation decreases as n increases. Furthermore, it can be shown that for $n = 2$ the orness value of this aggregation is always equal to α. This fact along with the anti-monotonicity with respect to n results in the notable feature that for any value of $n > 2$ the degree of orness is lower than the value of the parameter α. For this reason we shall call the OWA operators associated with these weights *pessimistic exponential OWA operators*. In particular, we note that as the number of arguments increases this aggregation becomes more and more andlike.

The optimistic and pessimistic exponential OWA operators have one very useful property. Given a value of n and a desired degree of orness one can simply obtain from Figs. 1 or 2 the associated value of α. Then the OWA weights can easily be generated according to (5) or (6). The following example demonstrates this simple method for generation of OWA weights.

Example 2. Let us assume the number of aggregate objects is 5. We shall construct the OWA weights that guarantee the desired degree of orness. Assume we desire a degree of orness of 0.6. From Fig. 2 we obtain the associated value of α as 0.8. By substituting $\alpha = 0.8$ in (6) we get the following OWA weights:

$$w_1 = 0.41;\quad w_2 = 0.10;\quad w_3 = 0.13;\quad w_4 = 0.16;\quad w_5 = 0.20.$$

For those OWA weights the exact degree of orness calculated by (1) is 0.5904. Similarly for a desired degree of orness 0.9 from Fig. 1 we obtain the associated with it the reading $\alpha = 0.7$. By the substitution in (5) we get the following OWA weights:

$$w_1 = 0.70;\quad w_2 = 0.21;\quad w_3 = 0.06;\quad w_4 = 0.02;\quad w_5 = 0.01.$$

The exact degree of orness for those weights is 0.8137.

This simple technique eliminates the need to calculate the OWA weights via the sophisticated procedure suggested in [5] which requires the solution of the constrained optimization problem. We are aware of the imprecision of the readings from the charts in Figs. 1 and 2. However, it is not reasonable to expect high accuracy in the expert estimation of the degree of orness. Therefore, such an approximate method for generation of the OWA weights is completely satisfactory for the solution of a large class of practical problems.

5. An alternative method for generation of OWA weights

We shall transform the formula for the degree of orness (1) in order to obtain an alternative expression for the relationship between the degree of orness and the OWA weights. Using the identity

$$\frac{1}{2}\sum_{i=1}^{n} w_i = \frac{1}{2}$$

we shall rewrite (1) as follows:

$$\operatorname{orness}(W) = \sum_{i=1}^{n} \frac{(n-i)}{n-1} w_i = \frac{1}{2} + \sum_{i=1}^{n} \frac{(n-i)}{n-1} w_i - \sum_{i=1}^{n} \frac{1}{2} w_i,$$

$$\operatorname{orness}(W) = \frac{1}{2} + \sum_{i=1}^{n} \left(\frac{(n-i)}{n-1} - \frac{1}{2}\right) w_i = \frac{1}{2} + \sum_{i=1}^{n} \frac{(n-2i+1)}{2(n-1)} w_i,$$

$$\operatorname{orness}(W) = \frac{1}{2} + \sum_{i=1}^{n} q_i w_i.$$

We now consider the case where n is even, $n = 2m$. We note that for $i = k$, where $k \leqslant m$ and $i = n + 1 - k$ we get

$$q_k = \frac{n - 2k + 1}{2(n-1)},$$

$$q_{n+1-k} = \frac{n - 2n - 2 + 2k + 1}{n-1} = \frac{-n + 2k - 1}{2(n-1)} = -q_k.$$

Thus

$$\text{orness}(W) = \frac{1}{2} + \sum_{k=1}^{m} q_k(w_k - w_{n+1-k}).$$

If n is odd, then $n = 2m + 1$ and we get

$$\text{orness}(W) = \frac{1}{2} + \sum_{k=1}^{m} q_k(w_k - w_{n+1-k}) + q_{m+1}w_{m+1}.$$

But

$$q_{m+1} = \frac{2m + 1 - 2(m+1) + 1}{2(n-1)} = 0$$

and hence again we get for the orness

$$\text{orness}(W) = \frac{1}{2} + \sum_{k=1}^{m} q_k(w_k - w_{n+1-k}).$$

From this expression directly follows a method for constructing OWA operators with weights that have predefined degree of orness.

Let us assume a given degree of orness Ω. We shall assume that all weights except the w_1 and w_n are equal. With this assumption the orness function becomes simply:

$$\text{orness}(W) = \tfrac{1}{2} + q_1(w_1 - w_n) = \tfrac{1}{2} + \tfrac{1}{2}(w_1 - w_n).$$

From the desired degree of orness Ω we get an explicit expression for the difference between the first and last weight:

$$w_1 - w_n = 2(\Omega - 0.5). \tag{7}$$

We can choose w_1 and w_n to be any numbers from the unit interval that satisfy the above condition. Then the remaining weights should be distributed equally to satisfy the requirement:

$$\sum_{i=1}^{n} w_i = 1.$$

Therefore, we get

$$w_i = \frac{1}{n-2}[1 - (w_1 + w_n)], \quad i = (2, 3, \ldots, n-1). \tag{8}$$

We shall illustrate this method by the following example.

Example 3. Assume $n = 4$ and the desired degree of orness is $\Omega = 0.8$. Then

$$w_1 - w_4 = 2(0.8 - 0.5) = 0.6.$$

One possible selection is

$$w_1 = 0.6; \quad w_4 = 0.$$

Then for the remaining weights we calculate according to (8):

$$w_3 = w_4 = \tfrac{1}{2}[1 - (0.6 + 0)] = 0.2.$$

By substitution in the original formulae for the degree of orness we can check that the OWA weights:

$$w_1 = 0.6; \quad w_2 = 0.2; \quad w_3 = 0.2; \quad w_4 = 0$$

have degree of orness 0.8.

However, we could choose

$$w_1 = 0.8; \quad w_4 = 0.2.$$

Then apparently the remaining weights should be

$$w_3 = w_4 = 0.$$

We now consider a slight modification of the above procedure.

1. Assume

$$\Delta = 2(\Omega - 0.5).$$

2. Let

$$L = \frac{1}{n}(1 - |\Delta|).$$

3. For $i = 2, \dots, n-1$

$$w_i = L.$$

4. If $\Delta > 0$ then

$$w_1 = L + \Delta, \qquad w_n = L.$$

if $\Delta \leqslant 0$ then

$$w_1 = L, \qquad w_n = L + \Delta.$$

Consider the situation $\Delta > 0$, here we get $w_1 = L + \Delta$ and $w_i = L$ for $i = 2, \dots, n$. In this case we obtain for the aggregated value

$$F(a_1, \dots, a_n) = \Delta \operatorname*{Max}_i[a_i] + L \sum_{i=1}^{n} a_i = \Delta \operatorname*{Max}_i[a_i] + \frac{(1-\Delta)}{n} \sum_{i=1}^{n} a_i.$$

Thus

$$F(a_1, \dots, a_n) = \Delta \operatorname*{Max}_i[a_i] + (1 - \Delta)\operatorname{Ave}(a_1, \dots, a_n).$$

The above form is introduced in [12] which was called the S-OWA operator. We note that if $\Delta \leqslant 0$ we get

$$F(a_1, \dots, a_n) = \Delta \operatorname*{Min}_i[a_i] + (1-\Delta)\operatorname{Ave}(a_1, \dots, a_n),$$

which is also a form of S-OWA operator.

6. Conclusion

We derived a method for obtaining the OWA aggregating operator from a collection of samples with aggregated data. We developed computationally effective methods for calculating the weights of the OWA operator for an assigned degree of orness.

References

[1] R.G. Brown, *Smoothing, Forecasting and Prediction of Discrete Time Series* (Prentice-Hall, Englewood Cliffs, NJ, 1963).
[2] P. Eklund and F. Klawonn, Neural fuzzy logic programming, *IEEE Trans. Neural Networks* 3 (1992) 815–819.
[3] K.J. Engemann, H.E. Miller and R.R. Yager, Decision making with belief structures: an application in risk management, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 4 (1996) 1–26.
[4] J. Kacprzyk, Inductive learning from considerably erroneous examples with a specificity based stopping rule, *Proc. Internat. Conf. on Fuzzy Logic and Neural Networks*, Iizuka, Japan (1990) 819.
[5] M. O'Hagan, Aggregating template rule antecedents in real-time expert systems with fuzzy set logic, Proc. 22nd Ann. IEEE Asilomar Conf. on Signals, Systems and Computers, Pacific Grove, CA (1988) 681–689.
[6] T. Rubinson, Communication networks, Ph.D. Thesis, Polytechnic Institute of New York, 1992.
[7] R.R. Yager, On ordered weighted averaging aggregation operators in multi-criteria decision making, *IEEE Trans. Systems Man Cybernet.* **18** (1988) 183–190.
[8] R.R. Yager, OWA neurons: a new class of fuzzy neurons, Proc. Internat. Joint Conf. on Neural Networks, Vol. I, Baltimore, I (1992) 226–231.
[9] R.R. Yager, Decision making under Dempster–Shafer uncertainties, *Internat. J. Gen. Systems* **20** (1992) 233–245.
[10] R.R. Yager, Families of OWA operators, *Fuzzy Sets and Systems* **59** (1993) 125–148.
[11] R.R. Yager and D.P. Filev, Fuzzy logic controllers with flexible structures, *Proc. Second Internat. Conf. on Fuzzy Sets and Neural Networks*, Iizuka, Japan (1992) 317–320.
[12] R.R. Yager and D.P. Filev, Parameterized "andlike" and "orlike" OWA operators, *Internat. J. Gen. Systems* **22** (1994) 297–316.
[13] J.M. Zaruda, *Introduction to Artificial Neural Systems* (West Publishing Co. St Paul, MN, 1992).