A family of OWA operators based on Faulhaber’s formulas

O.G. Duarte\textsuperscript{a,}\textsuperscript{*}, S.M. Téllez\textsuperscript{a,}\textsuperscript{**}

\textsuperscript{a}Universidad Nacional de Colombia, Facultad de Ingeniería, Bogotá, Colombia

Abstract

In this paper we develop a new family of Ordered Weighted Averaging (OWA) operators. Weight vector is obtained from a desired orness of the operator. Using Faulhaber’s formulas we obtain direct and simple expressions for the weight vector without any iteration loop. With the exception of one weight, the remaining follow a straight line relation. As a result, a fast and robust algorithm is developed. The resulting weight vector is suboptimal according with the Maximum Entropy criterion, but it is very close to the optimal. Comparisons are done with other procedures.

Keywords: Data fusion, Aggregation operators, Ordered Weighted Averaging

1. Introduction

Ordered Weighted Averaging (OWA) operators are well known aggregation operators \cite{1} that have received great attention in recent years. Although OWA operators were proposed by Yager since 1988 \cite{3}, a great amount of research arises since 2008. \cite{9} provides a collection of recent developments around OWA operators. A systematic review of literature about OWA is done in \cite{2}; it reveals some great research lines, one of them is to solve the weight vector determination problem. Another wide area of research deals with generalizations to fuzzy set theory.

The weight vector determines the way in which the OWA operator aggregates information. For some applications we need to adjust the behavior of the OWA operator to resemble the Minimum or Maximum operators. Those operators are numerical implementations of logical operators AND and OR. To handle that, the orness of the OWA operator was defined by Yager as a measure of how close to the OR operator the OWA is.

A common problem is to determine the vector weight of an OWA operator that meets a desired orness. A lot of methods have been proposed to solve this problem \cite{7}. In this paper we propose another one. Our method is fast, robust,

\textsuperscript{*}Corresponding author

\textsuperscript{**}This work was partially supported by Colciencias, call number 617
has good performance and is easy to interpret by non-experts. Some of these features appear in some methods, and other features in others, but our method accomplishes all of them. As an example, consider the exponential method [3]: it is fast and robust, but its performance (measured with the entropy index) is not the best; our method is faster, as robust and performs better than the exponential. As another example, consider the Maximum Entropy method [5], whose performance is optimal, but it is slow and have numerical problems; our method is faster and more robust than it, and its performance is very close to the optimal.

The structure of the paper is the following: section 2 summarizes the background about OWA operators. In section 3 we propose a new method to obtain the weight vector of an OWA operator that meets a desired orness. We explore the behavior and results of the proposed method in section 4. Some conclusions are done in section 5.

2. OWA operators

OWA operators are aggregation operators defined by [3, 1]:

\[
    OWA : \mathbb{R}^n \to \mathbb{R} \\
    y = f(x_1, x_2, \cdots, x_n) \\
    y = \sum_{i=1}^{n} w_i \bar{x}_i \quad \sum_{i=1}^{n} w_i = 1 \quad 0 \leq w_i \leq 1 \quad \forall i
\]  

(1)

Where \( \bar{x}_i \) is the \( i \)-th element of \( \{x_1, x_2, \cdots, x_n\} \) previously ordered from highest to lowest. The selection of the weight vector \( w = \{w_1, \cdots, w_n\} \) establishes different kind of aggregations; remarkable OWA operators are:

**Maximum:** with \( w = \{1, 0, 0, \cdots, 0\} \)

**Minimum:** with \( w = \{0, 0, \cdots, 0, 1\} \)

**Simple average:** with \( w = \{1/n, 1/n, \cdots, 1/n\} \)

2.1. Orness and entropy

Yager define orness\((w)\) and \(\text{disp}(w)\) (dispersion or entropy) as:

\[
    \text{orness}(w) = \frac{1}{n-1} \sum_{i=1}^{n} (n-i) w_i \\
    \text{disp}(w) = -\sum_{i=1}^{n} w_i \ln w_i
\]  

(2)

Orness refers to the degree to which the aggregation is like an or operation where as dispersion measures the degree to which \( w \) takes into account all information in the aggregation.
2.2. Families of OWA operators

There are a lot of strategies to compute the weight vector of an OWA operator. An extensive review is found in [7] where they are classified in 5 groups:

1. Optimization methods. The problem is stated as the optimization of a performance index subject to restrictions. Shannon entropy, dispersion, Rényi entropy are some of the indexes used. One of the most remarkable method is the Maximum Entropy method proposed by O’Hagan [8] that was reformulated in [5] in an analytical way.

2. Empirical data methods. This group of methods are usually intended to model the risk attitude of a decision maker: from a set of recorded decisions we must find the OWA that better reflects them.

3. Regular Increasing Monotone (RIM) quantifier methods. RIMs are linguistic quantifiers that can be used to guide the selection of the weight vector.

4. The argument dependent methods. These are methods that use the input data vector $x$ as useful information in the weight vector determination problem. Unlike the classical OWA operators, inputs do not have to be ordered.

5. Preference relation methods. These methods are an extension of the empirical data methods. Empirical data is replaced by a preference matrix provided by the experts.

A sixth group of methods can be added. It can be called Shaped based methods, because they use a predefined shape of the weight distribution to solve the problem. Notable methods are the exponential proposed by Yager in [3] and the linear proposed by Lamata in [6].

3. A family of OWA operators

We now address the issue to find the weights of an OWA operator with a desired orness. The operator must aggregate the information of $n$ individual values. We define $\alpha \in [0, 1]$ as a variable that equals the desired orness. However, in a more general perspective, $\alpha = g(orness) \in [0, 1]$ with $g(\cdot)$ a monotonically increasing function with $g(0) = 0$ and $g(1) = 1$.

We impose the following restrictions, intended to construct a family whose behavior is easy to understand:

OWA.1 if $\alpha = 0.0$ the operator must be equivalent to the Minimum operator.

OWA.2 if $\alpha = 0.5$ the operator must be equivalent to the Simple Average operator.

OWA.3 if $\alpha = 1.0$ the operator must be equivalent to the Maximum operator.
operator’s behavior must be symmetrical respect to $\alpha = 0.5$. In other words, the way in which the operator evolves from Minimum to Simple average when $\alpha$ varies from 0.0 to 0.5 must be the same way in which it evolves from Maximum to Simple Average when $\alpha$ varies from 1.0 to 0.5.

Last restriction is easy to accomplish if we define a family for $0 \leq \alpha \leq 0.5$ and then we use symmetry to define the behavior for $0.5 < \alpha \leq 1.0$.

3.1. And-like operators ($0 \leq \alpha \leq 0.5$)

Be $\alpha$ the desired orness, and consider first the case $0 \leq \alpha \leq 0.5$. The rationale of our proposal is the following: as $\alpha$ varies from 0 to 0.5 the weights must vary from those shown in figure 1(a) to those in figure 1(b). We first vary $w_n$ monotonically from 1.0 to $1/n$. The amount in which we diminishes $w_n$ is distributed into the other weights to keep unchanged the sum of weights. We distribute this amount using a straight line whose parameters are easy to compute, as we show in the following paragraphs.

We define $f(x) : [0, 0.5] \rightarrow [0, 1]$ a monotonically increasing function with $f(0) = 0$ y $f(0.5) = 1$. Other conditions for $f(x)$ are developed later. The value of $w_n$ must vary from 1 to $1/n$ when $\alpha$ varies from 0 to 0.5. We use $f(\cdot)$:

$$w_n = 1 - \Delta \quad \Delta = f(\alpha)(1 - 1/n) = f(\alpha)(n - 1)/n$$

Under this conditions, the sum of the other weights must be

$$\sum_{i=1}^{n-1} w_i = \sum_{i=1}^{m} w_i = \Delta \quad m = n - 1$$

According with equation 2 the orness is

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

We propose a distribution of $\Delta$ following the linear relation:

$$w_i = Ki + b \quad i = 1, 2, \cdots, m$$

Notice that this proposal is different from the linear proposal of Lamata [6], because we use the linear relation for $n - 1$ weights and Lamata uses it for all the $n$ weights.

Equations 5 and 6 turns into

$$\begin{cases} \sum_{i=1}^{m} (K_i + b) = \Delta \\ \frac{1}{m} \sum_{i=1}^{m} (n - i) (K_i + b) = \alpha \end{cases}$$
That can be arranged as

\[
\begin{cases}
K \sum_{i=1}^{m} i + \sum_{i=1}^{m} b = \Delta \\
\frac{K}{m} (n \sum_{i=1}^{m} i - \sum_{i=1}^{m} i^2) + \frac{b}{m} (\sum_{i=1}^{m} n - \sum_{i=1}^{m} i) = \alpha
\end{cases}
\]  

(9)

We compute the sums using Faulhaber’s formulas

\[
\begin{align*}
\sum_{i=1}^{m} i &= \frac{m(m + 1)}{2} \\
\sum_{i=1}^{m} i^2 &= \frac{m(m + 1)(2m + 1)}{6}
\end{align*}
\]

\[
\begin{cases}
K \frac{m(m+1)}{2} + mb = \Delta \\
K \frac{m(m+1)(m+2)}{6} + \frac{b}{m}(m+1)\frac{m+1}{2} = \alpha
\end{cases}
\]  

(10)

And rearrange the terms in order to show explicitly a system of 2 linear equations and 2 unknowns

\[
\begin{cases}
K \frac{m(m+1)}{2} + bm = \Delta \\
K \frac{(m+1)(m+2)}{6} + \frac{b}{m} \frac{m+1}{2} = \alpha
\end{cases}
\]  

(11)

From (11) we get direct expressions for \(K\) and \(b\)

\[
\begin{cases}
K = 6\frac{\Delta n/m - 2\alpha}{m^2-1} \\
\frac{b}{m} = \frac{\Delta}{m} - K \frac{(m+1)}{2}
\end{cases}
\]  

(12)

Using equation (11) we can compute \(K\) and \(b\) from \(\alpha\) and \(m\) or \(n\)

\[
\begin{cases}
K = 6\frac{f(\alpha) - 2\alpha}{m^2-1} = 6\frac{f(\alpha) - 2\alpha}{n(n-2)} \\
b = \frac{f(\alpha)}{m+1} - K \frac{(m+1)}{2} = \frac{f(\alpha)}{n} - K \frac{m}{2}
\end{cases}
\]  

(13)

5
3.2. Selection of $f(\cdot)$

As all weights must be non-negative, let's consider the first weight $w_1 = K \times 1 + b$

\[
K + b \geq 0
\]

\[
K + \frac{\Delta}{m} - K \frac{m+1}{2} \geq 0
\]

\[
K \frac{1-m}{2} + \frac{\Delta}{m} \geq 0
\]

\[
K \frac{1-m}{2} \geq -\frac{\Delta}{m}
\]

\[
3(1-m) \frac{\Delta n/m - 2\alpha}{m^2 - 1} \geq -\frac{\Delta}{m}
\]

\[
-\frac{3}{n}(\Delta \frac{n}{m} - 2\alpha) \geq -\frac{\Delta}{m}
\]

\[
2m\alpha - n\Delta \geq -n\Delta/3
\]

\[
2m\alpha \geq 2n\Delta/3
\]

\[
3\alpha \geq \Delta n/m
\]

\[
3\alpha \geq f(\alpha)
\]

To ensure that the other weights are also non-negative, we impose the condition $K \geq 0$. Directly from equation 13 we get $f(\alpha) \geq 2\alpha$.

We have found the following restrictions for $f(\alpha)$:

\[
2\alpha \leq f(\alpha) \leq 3\alpha
\]

If we choose $f(\alpha) = 1 - (1 - 2\alpha)^\beta$ with some $0 \leq \beta \leq 1$ (figure 2), those restrictions are satisfied if the slope at 0 is in $[2, 3]$

\[
2 \leq \frac{df(\alpha)}{d\alpha}|_{\alpha=0} \leq 3
\]

\[
2 \leq 2\beta(1 - 2\alpha)^{\beta-1}|_{\alpha=0} \leq 3
\]

\[
2 \leq 2\beta \leq 3
\]

\[
1.0 \leq \beta \leq 1.5
\]

Notice that the limit case with $\beta = 1.0$ makes $f(\alpha) = 2\alpha$. It makes $K = 0$ and $b = 1/n$ meaning that $\Delta$ is homogeneously distributed between $w_1, \ldots, w_{n-1}$. 

6
3.3. Or-like operators \((0.5 \leq \alpha \leq 1)\)

To ensure a symetrical behaivor of the family, we need to adapt the previous result when \(0.5 \leq \alpha \leq 1\) in the following way:

- Compute the weights for \(\hat{\alpha} = 1 - \alpha\)
- Reverse the order of the weights.

Algorithm 1 implements the above results in pseudocode.

4. Examples and analysis

In order to analyze the algorithm performance, we compare their results for 3 values of \(\beta\) (1.0, 1.25 and 1.5) against two well known algorithms: exponential method and maximum entropy method. Exponential method is similar to our proposal in the sense that it also propose a shape for the distribution of \(\Delta\) (an exponential distribution); however, to fit the desired orness, \(\alpha\) must be preset according to pre-computed figures. Maximum entropy method has an optimization approach, and it involves the numerical solution of an equation.

Figure 3 shows the weights obtained for \(n = 5\) when the desired orness is 0.6. Notice the linear distribution from \(w_2\) to \(w_5\) in figure 3(a) and the Homogeneous distribution when \(\beta = 1.0\). Notice also that the linear method approaches better than the exponential the weight distribution of the optimal entropy method. In fact, the exponential method provides an increasing distribution whereas entropy and linear methods provide decreasing distributions.

Figure 4 shows how fast varies \(w_1\) when we change the desired orness for \(n = 5\). We have added the case of the exponential method without preset, in order to stress the presetting effect. Curves in figure 4(b) has a change when orness is 0.5, caused by the strategy adopted to ensure simetry. These figures also show that linear method behaves closer to entropy method than the exponential one.

Figure 5 shows the change in all the weights when we change the desired orness for \(n = 5\). It is important to remark the simetry of the entropy and linear methods. Similarity between figures 5(a) and 5(b) reinforces the idea that linear method with \(\beta = 1.5\) is very similar to maximum entropy method.

Figure 6 shows how varies dispersion (entropy) when we change the desired orness for \(n = 5\). Variations of entropy with the linear method for different \(\beta\) are minimal. Notice also that the entropy obtained by the linear method with \(\beta = 1.5\) is very close to the optimal obtained by the entropy method.

Table 1 shows a comparison of computing times. Algorithms have been implemented in OpenModelica. We have run every algorithm 20 times for \(n = 10\) and \(n = 100\). The average simulation times are in table 1 as relative times to the minimal average time. Exponential method, as described in [3], involves a graphical procedure for the preset of \(\alpha\); we have implemented it using an exhaustive search procedure, which explains the high times for this method. It is an unfair comparison, and we prefer to compare exponentials method times using an implementation without preseting \(\alpha\).
Table 1 shows that the linear method is the fastest one. It is faster than the entropy method, because it doesn’t need any iteration loop and it is faster than the exponential method because arithmetic operations are simpler.

Moreover, entropy method has important lacks (see remark 1 in [7]): objective function definition implies \( w_i > 0 \) and as Maximum and Minimum operators are OWA’s with most of the weights equal to zero, those cases can not be included. Even for \( \alpha \) close to 0 or 1 numerical problems may arise: Figure 7 shows the dispersion (entropy) and orness for the Maximum entropy method for desired orness \( \in [0.9, 1.0] \) obtained with our OpenModelica implementation. Notice the numerical problems that appear near \( \alpha = 0.98 \). Those numerical problems are not present in the linear method, because there are not iteration loops.

5. Conclusions

The linear method is a fast and reliable method for computing weights of OWA operators with a desired orness. It is as fast or faster than the exponential method. It is also more suitable for software implementations than the exponential method because does not use precomputed curves for presettings. It is more robust than optimization methods because there are no iterations loops. And it provides a suboptimal result, very close to the optimal.

References

Referencias

[1] Detyniecki, M., 2001. Fundamentals on Aggregation Operators. Computer Science division, University of California, Berkeley.

[2] Emrouznejad, A., Marra, M., 2014. Ordered weighted averaging operators 1988–2014: A citation-based literature survey. International Journal of Intelligent Systems 29 (11), 994–1014. URL http://dx.doi.org/10.1002/int.21673

[3] Filev, D., Yager, R. R., 1998. On the issue of obtaining owa operator weights. Fuzzy Sets and System 94, 157–169.

[4] Fritzson, P., 2014. Principles of Object Oriented Modeling and Simulation with Modelica 3.3. Wiley-IEEE Press.

[5] Fullér, R., Majlender, P., 2000. An analytic approach for obtaining maximal entropy owa operator weights. Tech. rep., Turku Centre for Computer Science.

[6] Lamata, M. T., Perez, E. C., 2009. Owa weights determination by means of linear functions. Mathware and soft computing 16 (2).
[7] Liu, X., 2011. Recent Developments in the Ordered Weighted Averaging Operators: Theory and Practice. Vol. 265 of Studies in Fuzziness and Soft Computing. Springer, Ch. A Review of the OWA Determination Methods: Classification and Some Extensions, pp. 60–90.

[8] O’Hagan, M., 1988. Aggregating template or rule antecedents in real-time expert systems with fuzzy set logic. In: Twenty-Second Asilomar Conference on Signals, Systems and Computers. Vol. 2. pp. 681–689.

[9] Yager, R. R., Kacprzyk, J., Beliakov, G. (Eds.), 2011. Recent Developments in the Ordered Weighted Averaging Operators: Theory and Practice. Vol. 265 of Studies in Fuzziness and Soft Computing. Springer.
input: $n$ the size of the OWA operator
input: $orness$ the desired orness of the OWA operator
output: $w$ a vector of size $n$ with the weights of the OWA operator

1. Define $f(x) = 1 - (1 - 2x)^\beta$ with $\beta \in [1, 1.5]$. By default $\beta = 1.5$
2. if $orness \leq 0.5$ then
   3. $\alpha \leftarrow orness$
   4. else
      5. $\alpha \leftarrow 1 - orness$
   end
7. $m \leftarrow n - 1$
8. $\Delta \leftarrow f(\alpha)m/n$
9. $K \leftarrow 6(f(\alpha) - 2\alpha)/(m^2 - 1)$
10. $b \leftarrow f(\alpha)/n - Kn/2$
11. if $orness \leq 0.5$ then
      12. $w_n \leftarrow 1 - \Delta$
      13. for $i \leftarrow 1$ to $m$ do
         14. $w_i \leftarrow Ki + b$
      end
16. else
      17. $w_1 \leftarrow 1 - \Delta$
      18. for $i \leftarrow 1$ to $m$ do
         19. $w_{i+1} \leftarrow K(n - i) + b$
      end
21. end

Algorithm 1: Algorithm to get the weights of an OWA operator using the linear method
Figura 1: Weights for 3 OWAs with \( n = 5 \)

\[ f(\alpha) = 1 - (2\alpha - 1)^\beta \] and its restrictions
Figura 3: Weights for $n = 5$ and desired orness 0.6.
Figura 4: Variation of $w_1$ for $n = 5$ as a function of the desired orness
Figura 5: Comparison of Weights for $n = 5$ as a function of desired Orness
Figura 6: Comparison of Orness and Dispersion (Entropy) for $n = 5$ as a function of desired Orness
Figura 7: Dispersion (Entropy) and Real Orness for the Maximum entropy method with desired orness close to Maximum, for $n = 100$
Cuadro 1: Relative computing times. Averages from 20 samples

| Method                        | $n = 10$ | $n = 100$ |
|-------------------------------|---------|-----------|
| Maximum entropy               | 2.91    | 2.26      |
| Exponential with preset       | 108.81  | 366.22    |
| Exponential without preset    | 1.05    | 2.04      |
| Linear with $\beta = 1.00$   | 1.05    | 1.01      |
| Linear with $\beta = 1.25$   | 1.03    | 1.02      |
| Linear with $\beta = 1.50$   | 1.00    | 1.00      |