Data Fusion

Instructor: Dr. Moshiri

Reyhane Vahedi 810101303

Spring 2024

Homework 1

Section 1:

(a) **Uncertain OWA**:

The basis of this method is similar to other OWA methods, however the difference is that this method represents measurements as mathematical intervals instead of exact values. Following the name of this method, these intervals indicate the uncertainty of measurements. The goal here is to determine optimal values for the problem parameters considering its innate uncertainty. This operator maps each observation or iteration of measurements, which has a dimension of (n) representing the number of test sensors, to a one-dimensional scalar that is the result of combining all the measurements. This one-dimensional scalar provides us with an output for decision-making. Now considering that the n test sensors and m iterations, and also having in mind that the measurements are represented as intervals and not scalars, the formulation of the problem according to the given paper is as followed. First, we need to formulate the intervals that we will be working with. The output a of each sensor's measurement is given in equation 1. In this phrase it is depicted that if the minimum and the maximum of the interval are equal, then the interval would be represented as a single scalar value and the uncertainty of measurement would be non-existent.

(1)

$$a = [a^L, a^U] = \{x \,|\, a^L \leq x \leq a^U\}$$

In other cases, there will be a need of comparing different intervals as outputs of different sensors or as different measurements in order to build the matrix B. This matrix in all OWA methods, consists of all sensor measurements sorted from largest to smallest in different rows for different iterations. For this purpose, this paper employs a probability based comparison method to compare and sort various intervals, or various outputs. This method is shown in equation 2 for the probability of output a being bigger than output b, and in equation 3 for the probability of output b being bigger than output a.

(2)

$$p(a \geq b) = \max\left\{1 - \max\left(\frac{b^U - a^L}{I_a + I_b}, 0\right), 0\right\}$$

(3)

$$p(a \geq b) = \max\left\{1 - \max\left(\frac{a^U - b^L}{I_a + I_b}, 0\right), 0\right\}$$

Also, this paper provides a probability based solution for comparing outputs a_i and a_j in the same iteration, which is given in equation 4.

(4)

$$p_i j = p(a_i \geq a_j) = \max\left\{1 - \max\left(\frac{a_j^U - a_i^L}{I_{aj} + I_{ai}}, 0\right), 0\right\}$$

Considering the previous equations, for each iteration, considering all measurements, we can build a matrix p, which consists of various $p_i j$ values as its elements. This matrix, is of size n*n and it can be seen in equation 5.

(5)

$$p = \begin{bmatrix} p11 & p12 & \dots & p1n \\ p21 & p22 & \dots & p2n \\ \vdots & \vdots & \ddots & \vdots \\ pn1 & pn2 & \dots & pnn \end{bmatrix}$$

In the previous matrix, it should be noted that $p_ij = 1/2$, $p_ij + p_ji = 1$, and $p_ij \geq 0$. Now with all these in mind, we can calculate the probability of interval a_i being larger than all other output intervals of the same iteration. (6)

$$p_i = \sum_{j=1}^{n} p_{ij} \quad \text{for} \quad i \in \mathbb{N}$$

Therefore, to sort the a_i's based on their probabilities, or p_i, from highest to lowest, we create matrix B for each observation. The remaining rows of matrix B are obtained in a similar manner by applying the same process and calculating the probabilities for all sensor measurements of each iteration.

Then the paper goes on to introduce the interval s_k, which represents the acceptable interval or aggregated value of our measurements, where the index k indicates the number of our observation. According to our initial assumption about the number of observations or number of experiments, k will be a natural number between 1 and m. We extend this concept to values of a_i as well and rewrite them as a_ki, where k represents the observation number or iteration number. With these definitions, we have:

(7)

$$a = [a_{ki}^L, a_{ki}^U] \quad s = [s_{ki}^L, s_{ki}^U] \quad i \in \mathbb{N}, \text{ and } k = 1, 2, \ldots, m$$

And sequentially the weights would be as followed in equation (8).

(8)

$$v = \left(\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \middle| 0 \leq \alpha_i \leq v_i \leq \beta_i \leq 1, \, i \in \mathbb{N}, \sum_{i=1}^{n} \alpha_i \leq 1, \sum_{i=1}^{n} \beta_i \geq 1 \right)$$

From here on, we will need to employ optimization techniques and decide on a cost function for calculating the weights. It is obvious that in order to achieve the acceptable interval condition for a measurement output, the operator output needs to be in the S_k interval. This condition is formulated as equation 9.

(9)

$$g(a_{k1}, a_{k2}, \ldots, a_{kn}) = s_k \quad \text{for} \quad k = 1, 2, \ldots, m$$

From the previous equation, it can be understood that the function g can be rewritten as a linear function of b_i values , which are the sorted versions of the a_i values according to the previous formulas. The coefficients of each b_i will represent the weights, which we will denote as v_i. In the following equation, the index k represents the iteration number.

(10)

$$s_k = \sum_{j=1}^{n} v_j b_{kj} \quad \text{for} \quad k = 1, 2, \ldots, m$$

According to the intervals that we considered for the b_i values, which were the sorted version of the a_i values, the following phrases can be deduced by considering the minimum and maximum of the intervals.

(11)

$$s_k^L = \sum_{j=1}^{n} v_j b_{kj}^L \quad s_k^U = \sum_{j=1}^{n} v_j b_{kj}^U \quad \text{for} \quad k = 1, 2, \ldots, m$$

Finally, we can formulate the optimization problem for calculating the weights by introducing the minimum error and using it as a cost function. The minimum error function here would be the difference between the operator outputs and the acceptable interval s.

(12)

$$\min e_{1k} = |\sum_{j=1}^{n} v_j b_{kj}^L - s_k^L| \quad \text{for} \quad k = 1, 2, \ldots, m$$

(13)

$$\min e_{2k} = |\sum_{j=1}^{n} v_j b_{kj}^U - s_k^U| \quad \text{for} \quad k = 1, 2, \ldots, m$$

(14)

$$\text{s.t.} \quad \alpha_i \leq v_i \leq \beta_i, \quad j \in \mathbb{N}, \quad \sum_{j=1}^{n} v_j = 1$$

The paper then solves this optimization problem and presents bounds for the two error factors e_{1k} and e_{2k}, while using - and + signs to show the higher and lower deviations of these bounds, with e_1's deviation being calculated from the minimum of the interval s and e_2's deviation being calculated from the maximum of the interval s.

In summary, the UOWA operator can manage various types of vague or undefined data and thus is a flexible method. Also, this type of operators can accumulate uncertain data into a singular decision while using non-linearity as a way of modeling complexity. On the other hand, these operators can lack interpret-ability and turn out to be too complex to compute, specially when dealing with large amounts of data. These operators have a high sensitivity to the choice of parameters as well which can be troubling when kick-starting such algorithms.

(b) **Induced OWA**:

As previously stated, the OWA aggregation is $F_w(a_1, \ldots, a_n) = W^T B$. By now, we know that for implementing this aggregation, ordering the arguments is the initial step to take, and it produces the ordered vector B. The difference of induced OWA is in this step, where we induce an order instead of sorting normally like we do in the basic case of OWA. In this method, firstly we consider that we have n arguments, a_1 to a_n that we want to aggregate using the OWA weighting vector W. Then, it should be noted that this method denotes each a_i value as a two-tuple $< u_i, a_i >$, which then will be called an OWA pair. Induced OWA, which is generally a more general approach to OWA aggregation, we should order the arguments from vector B, based on the u_i values of each tuple.

(15)

$$F_w(< u_1, a_1 >, \ldots, < u_n, a_n >) = W^T B_u$$

More specifically, the procedure that we need to take for calculating the OWA aggregation of the OWA pairs, shown in equation 15, is as follows. First we need to form the ordered argument vector B_u so that b_j is the a, or the secondary, value of the OWA pair having the j_{th} largest u, or primary, value among all tuples. In discussing these OWA pairs, or $< u_i, a_i >$, because of its role we shall refer to the u_i as the order inducing variable and a_i as the argument variable. Using this method for ordering our pairs, we can then proceed to add the pairs and their respective weights into equation 15, considering their order of magnitude, and yield the induced OWA aggregation.

In summary, the IOWA operators are communicative, meaning that the initial indexing of the pairs is unimportant, and they show bounding properties just like basic OWA operators do, meaning that for any order inducing and weighting vector, equation 16 stands.

(16)

$$min_j[a_j] \le F_w(< u_i, a_i >) \le max_j[a_j]$$

Other characteristics of these operators include idempotency and monotonicity. As a result of these characteristics, the IOWA operators can be seen as mean aggregation operators with respect to the argument variable. To list some of the pros of these operators, we can count their flexibility in decision making, aggregation of individual preferences(eg., from different expert opinions) into a collective decision, and their versatility. The cons of these operators on the other hand include, their subjectivity and lack of universal standards, and also their complexity of calculations alongside their sensitivity to weights assignments.

(c) **Linguistic OWA**:

In this method, instead of dealing with numerical values provided by the sensors as outputs, we work with linguistic concepts. These operators are used when even though the values at hand were precise, they would not be able to convey explicit concepts. Therefore, in such cases, linguistic concepts or linguistic terms can be used, and for each element specified in a given set, a membership function is assigned. (For example, in the article, this function is defined with four specific numbers, but membership functions can be represented in various ways.) As per said, the evaluation of a linguistic variable is based on its membership function. In this article, to represent the membership of a linguistic concept, the quadruple $(a_i, b_i, \alpha_i, \beta_i)$ is considered, where the first two numbers indicate the intervals in which this linguistic concept has a value of one in the membership function of the set, and the next two numbers, from left to right, indicate the left and right widths of membership in the membership function of this linguistic variable. For a better understanding of this concept, consider With these preparations, we can define the LOWA aggregation as per equation 17.

(17)

$$\phi(a_i, \ldots, a_m) = W.B^T = C^m(\omega_k, b_k, k = 1, \ldots, m) =$$
$$= \omega_1 \bigcirc b_1 \bigoplus (1 - \omega_1) \bigcirc C^{m-1}(\beta_h, b_h, h = 2, \ldots, m)$$

In equation 17, C is a convex combination operator that is defined in 18 and k in 18 is defined in 19. T is the number of linguistic operator labels and w_1 is the first weight of the operator, belonging to the largest element.

(18)

$$C^2(\omega_i, b_i, i = 1, 2) = \omega_1 \bigcirc s_j \bigoplus (1 - \omega_1) \bigcirc s_i = s_k, (i \le j)$$

(19)

$$k = min(T, i + round(w_1.(j - i)))$$

In summary, some of the pros of the LOWA operators would be their flexibility and versatility in being used in various fields and scenarios. These operators are also interpretable and can be set to fit very personal cases and usages. The LOWA operators are also resistant to noise and can be used in real life applications. The cons of using these operators would be that they are complex in implementation, specially when dealing with large amounts of data, they are biased towards the users interpretations, they can have limited use since they are conceptual, and that there is no standard approach to learning the weights of these operators which can consequently make it hard to compare the results of different LOWAs.

(d) **Dependent OWA**:

In this paper the focus is on the step of weight learning rather than ordering. This paper classifies all previously introduced operators into two categories. The weights derived by the "argument-independent" approaches are associated with particular ordered positions of the aggregated arguments, and have no connection with the aggregated arguments, while the "argument-dependent" approaches determine the weights based on the input arguments. In this paper, the second category is discussed, and a new argument-dependent approach to determining the OWA weights is introduced.

In many actual situations, the arguments $a_1, a_2, \ldots, a_n$ are usually given by n different individuals. Some individuals may provide unduly high or unduly low preference arguments for their preferred or repugnant objects. In these cases, we would assign very low weights to these "false" or "biased" opinions, that is to say, the closer a preference argument to the average value, the more the weight. This is the issue that gives rise to the introduced dependent OWA weight calculating method of this paper. Considering arguments a_1 to a_n, and an average value of all arguments, or μ, then $(\sigma(1), \sigma(2), \ldots, \sigma(n))$ would be a permutation of (1, 2, ..., n), such that $\alpha_{\sigma(j-1)} \geq alpha_{\sigma(j)}$, for all $j = 1, \ldots, n$, and $s(a_{\sigma(j)}, \mu)$ would be the similarity degree of the j_{th} largest argument $a_{\sigma(j)}$ and the average value μ. Then if $s(\alpha_{\sigma(i)}, \mu) \geq s(\alpha_{\sigma(j)}, \mu)$, $w_i \leq w_j$ would stand.

Also, considering the n arguments, if $a_i = a_j$, for all i and j, then $w_j = 1/n$ for all j.

In summary, this dependent OWA operator can relieve the influences of unfair arguments on the aggregated results, and it is practical and effective in calculating optimal weights.

Section 2:

Here by using the given hint, we calculated the matrix p in 2.1, and summed up each row of p in 2.2 to find out the order of intervals. lastly the corrected order of intervals can be seen in 2.3.

(2.1)

$$p_{ij} = \begin{bmatrix} 1/2 & 2/3 & 1/3 & 1 \\ 1/3 & 1/2 & 1/4 & 2/3 \\ 2/3 & 3/4 & 1/2 & 1 \\ 0 & 2/3 & 0 & 1/2 \end{bmatrix},$$

(2.2)

$$p_1 = 2.5, p_2 = 1.75, p_3 = 1.9167, p_4 = 1.1667$$

(2.3)

the right order of intervals:a_1->a_3->a_2->a_4

Section 3:

(a) Based on the IOWA paper, there is a distinct difference between the aggregation of OWA pairs of the Induced OWA operator formulation and the ordinary OWA aggregation. This distinction arises when there is a tie in the ordering operation, or as this question states, when $u_i = u_j$. In this case there will be a tie between the two intervals starting with u_i and u_j, and they would be ordered on a single step consisting both of these intervals. When such a case happens, choosing either the tied intervals before the other would yield a different ordered argument vector, B. (This vector, as previously stated, is made of the $\tilde{s_i}$ elements of all the intervals) The proposed policy for dealing with this problem is to replace the arguments of the tied pairs by the average of the arguments of the tied pairs in forming the B vector. For this specific problem I think that employing the same technique for averaging the α indices of the s elements would solve the ordering error, just the same as the previously explained solution.

(b) According to the given phrase explaining our operator, and also the inducing OWA algorithm, we can write expression 3.1 to be able to interpret a linguistic result:

(3.1)

$$\text{IU-OWA}(\langle 0.6, [s_{-1}, s_1]\rangle, \langle 0.2, [s_{-2}, s_1]\rangle, \langle 0.8, [s_1, s_2]\rangle, \langle 0.4, [s_1, s_3]\rangle) =$$

$$0.2 \times [s_1, s_2] \oplus 0.3 \times [s_{-1}, s_1] \oplus 0.1 \times [s_1, s_3] \oplus 0.4 \times [s_{-2}, s_1] =$$

$$[s_{0.2}, s_{0.4}] \oplus [s_{-0.3}, s_{0.3}] \oplus [s_{0.1}, s_{0.3}] \oplus [s_{-0.8}, s_{0.4}] = [s_{-0.8}, s_{1.4}]$$

It can be seen that the acquired interval consists of the values of s_0 and s_1 which linguistically means that the decision can either be good or average.

(c)First, for ordering such pairs, we will need to calculate the matrix p of probabilities and use each element of it to decide which $\tilde{s}$, or u_i, is larger than the other. Following the explanations given in question 1, and also the hints given in question 2, the following equations can be used for the calculation and comparison of probabilities:

(3.2) for $\tilde{s_i} = [s_{\alpha 1}, s_{\beta 1}]$, $\tilde{s_j} = [s_{\alpha 2}, s_{\beta 2}]$:

$$p_{ij} = \frac{\max\{0, \text{len}(\tilde{s_i}) + \text{len}(\tilde{s_j}) - \max\{\beta_2 - \alpha_1, 0\}\}}{\text{len}(\tilde{s_i}) + \text{len}(\tilde{s_j})}$$

$$p_{ij} = p(\tilde{s_i} \geq \tilde{s_j}) -> p = \begin{bmatrix} p11 & p12 & \dots & p1n \\ p21 & p22 & \dots & p2n \\ \vdots & \vdots & \ddots & \vdots \\ pn1 & pn2 & \dots & pnn \end{bmatrix}, p_{ij} \geq 0, p_{ij} + p_{ji} = 1, p_{ii} = 1/2$$

$$\tilde{s_i} -> p_i = \sum_{j=1}^{n} p_{ij}$$

Using the equations of (3.2) the matrix p can is yielded and can be seen in 3.3. For yielding this matrix, we considered all the first intervals of the four given interval pairs. For easier reference, we rename the interval pairs as $[\tilde{s_i}', \tilde{s_i}]$ and proceed to compare all $\tilde{s_i}'$s of the pairs, thus the p matrix will be of size 4*4. (3.3)

$$p_{ij} = \begin{bmatrix} 1/2 & 2/3 & 3/4 & 1/2 \\ 1/3 & 1/2 & 3/5 & 1/3 \\ 1/4 & 2/5 & 1/2 & 1/4 \\ 1/2 & 2/3 & 3/4 & 1/2 \end{bmatrix},$$

also, from each row of the matrix we have: (3.4)

$$p_1 = 2.4167, p_2 = 1.7667, p_3 = 1.4, p_4 = 2.4167$$

According to 3.4, the ordering of the interval pairs per $\tilde{s_i}'$ will be as shown in 3.5. However, it can be seen that since we have two similar $\tilde{s_i}'$s, two pairs will be ordered in one level, this problem can be solved by using the method explained in part (a) and averaging the corresponding $\tilde{s_i}$s of the equal $\tilde{s_i}'$s and replacing the two intervals, $[s_{-1}, s_1$ and $[s_1, s_3]$, with their equivalent average or $\frac{\tilde{s_i} \oplus \tilde{s_j}}{2}$ which is $[s_0, s_{1.5}]$.

(3.5)

$$< [s_0, s_1], [s_0, s_{1.5}] >, < [s_0, s_1], [s_0, s_{1.5}] >, < [s_{-1}, s_1], [s_{-2}, s_1] >, < [s_{-2}, s_1], [s_1, s_2] >,$$

Then we can proceed to employ the IU-OWA operator as we previously did in part (b):

(3.6)

$$\text{IU-OWA}(\langle [s_0, s_1], [s_{-1}, s_1]\rangle, \langle [s_{-1}, s_1], [s_{-2}, s_1]\rangle, \langle [s_{-2}, s_1], [s_1, s_2]\rangle, \langle [s_1, s_3], [s_1, s_3]\rangle) =$$

$$0.2 \times [s_0, s_{1.5}] \oplus 0.3 \times [s_0, s_{1.5}] \oplus 0.1 \times [s_{-2}, s_1] \oplus 0.4 \times [s_1, s_2] =$$

$$[s_0, s_{0.3}] \oplus [s_0, s_{0.45}] \oplus [s_{-0.2}, s_{0.1}] \oplus [s_{0.4}, s_{0.8}] = [s_{0.2}, s_{1.65}]$$

From a linguistic point of view, this yielded interval can be interpreted as "good" since the only index of s that is included in [0.2, 1.65] is 1.

Section 4:

Using the given function of question 3, here we will first use the b weights and apply the operator to each column of each table, thus yielding a single interval per a_i for each expert. (It should be noted that I added 0.1 to the third weight in order to fix the issue with the sum of weights, so I will be using $w_b^T = [0.3, 0.2, 0.2, 0.3] through this section$) One of the needed calculations has been shown in 4.1, the rest have been done and shown in tables 1 and 2. (4.1) For the first expert:

IU-OWA of a_1 for b_1 to $b_4([s_1, s_2], [s_1, s_3], [s_2, s_3], [s_0, s_2] = 0.3\times[s_1, s_2]\oplus0.2\times[s_1, s_3]\oplus0.2\times[s_2, s_3]\oplus0.3\times[s_0, s_2] = [s_{0.3}, s_{0.6}]\oplus$

Table 1: Expert E_1

a_1	a_2	a_3	a_4
$[s_{0.9}, s_{2.4}]$	$[s_{-0.8}, s_{1.2}]$	$[s_{0.7}, s_2]$	$[s_{0.9}, s_{2.1}]$

Table 2: Expert E_2

a_1	a_2	a_3	a_4
$[s_{0.2}, s_{1.5}]$	$[s_1, s_2]$	$[s_{0.4}, s_{1.8}]$	$[s_{0.9}, s_{2.4}]$

Then we need to repeat the same process, but with the weights assigned to the experts' opinions. More specifically, we need to apply the weights w_{E_i} and apply the operator to the two intervals we have for each a_i. The calculations for the first interval using w_E has been shown in 4.2 and the results are given in table 3.

(4.2)

$$\text{IU-OWA of } a_1 \text{ for } E_1 \text{ and } E_2([s_{0.9}, s_{2.4}], [s_{0.2}, s_{1.5}]) = 0.6\times[s_{0.9}, s_{2.4}]\oplus0.4\times[s_{0.2}, s_{1.5}] = [s_{0.54}, s_{1.44}]\oplus[s_{0.08}, s_{0.6}]$$

$$= [s_{0.62}, s_{2.04}]$$

Table 3: Final a_is

a_1	a_2	a_3	a_4
$[s_{0.62}, s_{2.04}]$	$[s_{-0.8}, s_{1.52}]$	$[s_{0.66}, s_{1.92}]$	$[s_{0.9}, s_{2.22}]$

Lastly, without any need to calculate the p matrix, it can be seen that a_1 and a_4 both include the "good" and "very good" levels, which would be better than decisions a_2 and a_3, with only "average" and "good" levels respectively. Between a_4 and a_1, a_4 would be a better decision with a lower uncertainty than a_1, considering their respective lengths of 1.32 and 1.42.

Section 5:

The codes for this problem have been attached to this report.

(a) **Sorting the sensors per precision by using various error metrics:**

The asked error rates are reported in the table shown in figure 2. From this table, it can bee seen that these sensors have relatively similar MAE and MSE values but different error variances. So by considering the last column as a measure of sensor precision, we can say that the third sensor acts better than the first, which acts better than the second sensor.

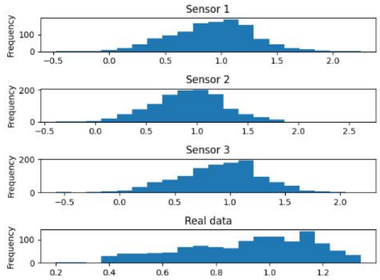

Figure 1: histograms of all sensors and real data

data	MAE	MSE	Error Variance
sensor 1	0.2038	0.0783	0.0470
sensor 2	0.2057	0.0781	0.0836
sensor 3	0.2093	0.0789	0.0003

Figure 2: MAE, MSE, and Error variance of the three sensors

(b) **Comparing different weight learning approaches:**
(b.i) **Optimistic Approach**:

Weight	Value
W1	0.3
W2	0.21
W3	0.49

Figure 3: Weights of optimistic approach, $\alpha = 0.3$

error metric	Value
MAE Optimistic	0.2993
MSE Optimistic	0.1452

Figure 4: Error metrics of optimistic approach, $\alpha = 0.3$

metric	Value
orness	0.405
dispersion	1.0384

Figure 5: Orness and dispersion of the optimistic approach, $\alpha = 0.3$

(b.ii) **Pessimistic Approach**:

Pessimistic Weight	Value
W1	0.09
W2	0.21
W3	0.7

Figure 6: Weights of pessimistic approach, $\alpha = 0.3$

error metric	Value
MAE pessimistic	0.3154
MSE pessimistic	0.1620

Figure 7: Error metrics of pessimistic approach, $\alpha = 0.3$

metric	Value
pessimistic orness	0.195
pessimistic dispersion	0.7941

Figure 8: Orness and dispersion of the pessimistic approach, $\alpha = 0.3$

(b.iii) **Induced OWA Approach**:

Induced Weight	Value
W1	0.4817
W2	0.2746
W3	0.2436

Figure 9: Weights of induced OWA approach, $\beta = 0.01$

Induced	value
Orness	[0.63125753]
Dispersion	[1.04510084]
MAE	0.2962257191169825
MSE	0.1403418149586015

Figure 10: Error metrics, orness and dispersion of induced OWA approach, $\beta = 0.01$