CASC PROJECT

Computational Aspects of Statistical Confidentiality

June 2004

Scientific papers on semantics and aggregation procedures for SDC of qualitative variables

Vicenç Torra

Institut d'Investigació en Intel·ligència Artificial - CSIC
e-mail: vtorra@iiia.csic.es http://www.iiia.csic.es/~vtorra

Deliverable 1.1-D14

V. Torra, microagregation for categorical variables: a median based approach, Lecture Notes in Computer Science 3050 (2004) 162-174 (Privacy in Statistical Databases, Barcelona, 2004).

J. Domingo-Ferrer, V. Torra, Median based aggregation operators for prototype construction in ordinal scales, Intl. J. of Intel. Syst., 18:6 (2003) 633-655.

J. Domingo-Ferrer, V. Torra, Extending microaggregation procedures using defuzzification methods for categorical variables, Proc. IEEE Symposium on Intelligent Systems (ISBN 0-7803-7601-3), Special Session on Intelligent Adaptive Systems, 44-49, Bulgaria, Setiembre, 2002.

V. Torra, OWA operators in data modeling and re-identification, IEEE Trans. on Fuzzy Systems, in press.

Microaggregation for categorical variables: a median based approach

Vicenç Torra

Institut d'Investigació en Intel·ligència Artificial,
Campus de Bellaterra, 08193 Bellaterra, Catalonia, Spain
vtorra@iiia.csic.es

Abstract. Microaggregation is a masking procedure used for protecting confidential data prior to their public release. This technique, that relies on clustering and aggregation techniques, is solely used for numerical data. In this work we introduce a microaggregation procedure for categorical variables. We describe the new masking method and we analyse the results it obtains according to some indices found in the literature. The method is compared with Top and Bottom Coding, Global recoding, Rank Swapping and PRAM.

Keywords: Privacy preserving data mining, Data protection, Masking methods, Clustering, Microaggregation, Categorical data

1 Introduction

Companies and Statistical Offices collect data from respondents (either individuals or companies) to extract rellevant information or to inform policy makers and researchers. However, the fulfillment of this goal has to be done assuring confidentiality and, thus, avoiding the disclosure of respondents' sensitive data. This is, disclosure risk should be minimized. Statistical disclosure control (SDC) – or Inference Control – studies tools and methods (namely, masking methods) to allow dissemination of data protecting confidentiality. Privacy preserving data mining [1] is a related field with similar goals. While the former is oriented to statistical databases, the latter is oriented to company proprietary information.

It is important to note that a straightforward manipulation of the data is not enough for avoiding disclosure because data has to maintain the so-called analytical validity [25]. This is, in short, that the analysis performed on the protected data has to lead to results similars to the ones obtained using the original data. In other words, information loss should be small. See [7] and [24] for a state of the art description of the field.

For the purpose of data confidentiality, a plethora of masking methods have been designed. A comprehensive description of the methods and their properties is given in [4] and [24]. See also [5] for a comparative analysis of the methods with respect to some indices for measuring information loss and disclosure risk. In [9], an up to date review of the methods currently in use by National Statistical Offices is given.

Masking methods can be classified according to different dimensions. In relation to this work, it is relevant to classify methods according to the type of variables. Then, methods for categorical (either nominal or ordinal) variables and methods for numerical (continuous) variables can be distinguished. Among the methods for numerical variables, we can distinguish: rank swapping, microaggregation and noise addition. These methods are currently in use [9] and have good performance [5] in relation to information loss and disclosure risk indices. For categorical data, existing methods include Rank swapping, Top and Bottom coding, recoding and PRAM (Post-Randomization Method).

A detailed analysis of the methods (see *e.g.* [4]) shows that some of the methods for numerical variables are not appliable to categorical data (and vice-versa). This is due to the intrinsic nature of the variables and the difficulties of translating some numerical functions (*e.g.* addition, averaging) into a categorical domain. Among those methods we find microaggregation.

From the operational point of view, microaggregation consists on obtaining a set of clusters (gathering similar respondents) and, then, replacing the original data by the averages of all the respondents in the corresponding cluster. In this way, the data for each respondent is protected. For avoiding discloure, clusters have to contain a minimum number of respondents (otherwise the average does not avoid the disclosure because an individual contributing to the cluster, or an external individual, can guess the value of another respondent). Difficulties on extending this approach to categorical data rely on clustering and aggregation and their suitability to deal with categorical variables.

Although, in general, the interest of translating masking methods from one scale to another is not clear, the case for microaggregation is different. Numerical microaggregation performs quite well with respect to the different existing indices for information loss and disclosure risk. Moreover, it is shown in [5] that it is the second best rated method for numerical data, just behind rank swapping. Therefore, it seems appropriate to consider a categorical microaggregation and whether this method can also lead to such similar good results.

In this work we introduce a categorical microaggregation method and we show that it outperforms other masking methods for the same type of data. The method is based on clustering techniques (see *e.g.* [16, 17] and on some aggregation operators, both for categorical data.

The structure of this work is as follows. In Section 2, we describe the basic elements we need latter on for defining our categorical approach. Then, in Section 3, we describe the categorical microaggregation procedure. It follows, in Section 4, a detailed analysis of the experiments performed to evaluate our approach. The work finishes in Section 5 with some conclusions and future work.

2 Preliminaries

This section is divided in two parts. We start with a short description of microaggregation. Then, we review some aggregation procedures that can be used for categorical data.

2.1 Microaggregation

As briefly described in the introduction, microaggregation can be operationally defined in terms of the following two steps:

Partition: The set of original records is partitioned into several clusters in such a way that records in the same cluster are *similar* to each other and so that the number of records in each record is at least k.

Aggregation: The average value (a kind of prototype) is computed for each cluster and used to replace the original values in the records. This is, each record is replaced by its prototype.

According to this description, an actual implementation of microaggregation requires a clustering method and an aggregation procedure. While for numerical data the main difficulty is on the clustering method (most clustering methods do not apply because they do not satisfy the constraint about the minimal number of records in each cluster), for the categorical data difficulties appear in both processes.

In fact, while, up to our knowledge, there is no microaggregation method for categorical data, there exist several methods for numerical data. Differences on the latter methods correspond to differences on the way clusters are built (modification of standard techniques, novel approaches using genetic algorithms with an appropriate fitness function, . . .), on the way a large set of variables is considered (repeatedly applying univariate microaggregation – microaggregation for a single variable, applying multivariate microaggregation – all the variables at once, . . .) or on the aggregation procedure. In relation to the aggregation procedure, while the most common method is the arithmetic mean, other procedures, as the median operator [18], have also been used.

2.2 Aggregation procedures for categorical data

At present, there exist several aggregation functions for categorical data. See *e.g.*, [26] for a recent survey on aggregation operators. Here we can distinguish between operators for nominal scales (where only equality can be used to compare elements) and ordinal scales (there is an ordering among the elements). In the case of nominal scales, the main operator is the plurality rule (mode or the voting procedure).

In the case of ordinal scales, operators can be classified, following [22], in three main classes. We review below these classes considering the ordinal scale $L = \{l_0, \cdots, l_R\}$ with a total order $\leq_L$ (defined as follows: $l_0 \leq_L l_1 \leq_L \cdots \leq_L l_R$).

1. *Explicit quantitative or fuzzy scales:* A mapping from L to a numerical (or fuzzy) scale (say, N) is assumed. Then, aggregation functions are defined in this underlying N scale. In some cases, this numerical scale is not given but inferred from additional knowledge about the ordinal scale (*e.g.* from a one-to-many negation function [20]). The operator in [22] follows this approach.

2. *Implicit numerical scale:* An implicit numerical scale underlying the ordinal scale is assumed. Operations on L are defined as operating on the underlying scale. The usual case is that each category l_i is dealt as the corresponding integer i. This is the case of Linguistic OWA [11] and Linguistic WOWA [21].
3. *Operating directly on categorical scales:* Operators stick to a purely ordinal scale and are based only on operators on this scale. This is the case of the median operator or the Sugeno integral [19]. These operators solely rely on $\leq_L$ (or to *minimum* and *maximum*). Other operators in this class (e.g., the ordinal weighted mean defined in [10]) are based on t-norms and t-conorms (two operators that can be defined axiomatically over ordinal scales).

Aggregation operators for categorical scales where reviewed and analyzed in [6]. Revision was focused on their application for prototype construction (a case similar to the one considered here). Note that the aggregation step in microaggregation can be understood as building a centroid (a representative) for each cluster. In short, results show that the most relevant aggregation method for ordinal scales is the median (simpler to use and with a straightforward meaning) but that this operator does not allow for compensation. Recall that in this setting compensation implies that the aggregation of some values $l_{k_i} \in L$ can be a value in L different to the l_{k_i} but in the interval $[\min l_{k_i}, \max l_{k_i}]$. Also, the standard definition does not include weights. Then, [6] introduced the CWM to consider weights and to allow for compensation.

In this operator, a set of data sources X are assumed to supply values a_i (formally speaking $a_i = f(x_i)$), $p(x_i)$ are the importances of the sources $x_i \in X$. Additionally, a function Q is used to distort the weights. The role of Q is to distort the weights so that a greater importance is assigned to smaller, larger or central values.

Definition 1. *Let* $\mathbf{p} : X \to D \subset \mathbb{R}$ *be a weighting vector, let* Q *be a non-decreasing fuzzy quantifier (a non-decreasing function* $Q : [0,1] \to [0,1]$ *with* $Q(0) = 0$ *and* $Q(1) = 1$*), then the* CWM *operator of dimension* N *(*$CWM_{\mathbf{p}} : L^N \to L$*) is defined as:*

$$CWM_{\mathbf{p}}(a_1, \cdots, a_N) = a \textit{ if and only if } acc^{iv}(a) > 0.5 \geq acc^{iv}(b)$$

where b *is the element previous to* a *in* L *(*$b = \max\{x | x \in L, x < a\}$*) and where* $acc^{iv}(a) = \sum_{b \leq a} acc'''(b)$*,* acc''' *is the WOW-weighting vector of* (L, acc'') *and* Q*,* $acc''(a) = acc'(a) / \sum_{b \in L} acc'(b)$ *with* acc' *defined as:*

$$acc'(a) = \min(\max_{b \leq a} acc(b), \max_{b \geq a} acc(b)) \tag{1}$$

and where $acc(a) = \sum_{f(x_j) = a} p(x_j)$*.*

Roughly speaking, acc accumulates the weight of each element in L, acc' makes this function convex (to allow for compensation) and acc'' normalizes so that it adds to one, acc''' is a manipulation of this function (through Q to distort the importance of certain elements) and, finally, using acc^{iv} the element that occupies the central position is selected.

The WOW-weighting vector used in the above definition was defined as follows:

Definition 2. *Let $(a_i, p_i)_{i=1,N}$ be a pair defined by a value and the importance of a_i expressed in a given domain $D \subset \mathbb{R}^+$, and let Q be a fuzzy non-decreasing quantifier. Then, the WOW-weighting vector $\omega = (\omega_1, \cdots, \omega_N)$ for $(a, \mathbf{p})$ and Q is defined as follows:*

$$\omega_i = Q\big(\frac{\sum_{j \leq i} p_{\sigma(j)}}{\sum_{j \in L} p_{\sigma(j)}}\big) - Q\big(\frac{\sum_{j < i} p_{\sigma(j)}}{\sum_{j \in L} p_{\sigma(j)}}\big)$$

where σ is a permutation as above such that $a_{\sigma(i-1)} \geq a_{\sigma(i)}$.

3 Proposed method

The proposed method for categorical microaggregation is based on the methods for clustering and aggregation defined in the previous section. The proposed algorithm is as follows:

```
procedure microaggregation (M: data matrix; NVar: int) is
I:= select variables to be microaggregated (M);
for i:=1 to $|I|$ step NVar do
    WS:= projection of M on variables (i ... max($|I|$,i+NVar-1));
    WS2:= only different records from WS;
    FR:= frequency of records (WS2, WS);
    NClust:= appropriate number of clusters (WS2, FR);
    hard k-means of (WS2, FR);
    aggregation and replacement (WS2, FR, M);
end for;
end procedure;
```

This is, first the variables to be microaggregated are selected from the data matrix M. Then, groups of NVar variables are built from M defining a working space (WS). Then, the WS is reduced (WS2) so that only different records are allowed. For each record in WS2, its frequency in WS is computed and stored in FR. This frequency is used by the program to estimate an *appropriate* value for NClust (the number of clusters) to be used in the clustering process. Then, the clustering algorithm is applied. Finally, the original values are replaced by the new ones (the centroids of the clusters).

Now, we describe in more detail some of the elements that appear in the algorithm above:

Clustering: Our clustering algorithm is based on the k-modes algorithm. This latter algorithm, designed for categorical data (see [12]), is inspired on the k-means algorithm (for numerical data). In short, the method obtains the optimal cluster through an iterative process consisting on the following steps: (i) for each cluster, representatives are computed; (ii) records are assigned to their nearest cluster.

The following three aspects have been considered in our implementation:

a) To bootstrap the process, an initial partition is needed. We build it at random.

b) To determine which is the nearest cluster of a record, a distance, defined as the summation of the distance between individual values, is used. Here, nominal and ordinal scales are differentiated. For variables on nominal scales, distance is defined as 1 when values are different and 0 when equal. In ordinal scales, distance is defined according to the position of the categories in the domain.

c) To compute the representatives, an aggregation method is used variable by variable. We use the plurality rule (mode or voting procedure) in nominal scales. In ordinal scales, three alternatives have been considered: mode (as for nominal scales), the CWM (as defined above) and a CWM where a is the selected element if and only if $acc^{iv}(a) > \beta \geq acc^{iv}(b)$ (for a β randomly selected).

d) To assure that all final micro-clusters have a desired cardinality, some elements are relocated.

Aggregation: For aggregation, we apply the same process used for computing cluster representatives in the clustering algorithm.

4 Results

In this section we describe the results obtained for our masking method. We start describing the methodology used to evaluate our method and then the experiments and the conclusions of them.

4.1 Evaluation method

To evaluate our approach we have applied the methodology previously used in [5] and in [27] consisting in developing a score combining two measures, one for disclosure risk and the other for information loss. The score can be computed for any pair (original-file, masked-file). Then, a data file was masked using different masking methods (and considering different parameterizations for each masking method) and the scores for each pair (original-file, masked-file) were obtained and compared.

According to this, we got a score for each pair (masking-method, parameterization). Now we consider the computation of the score and the masking methods we have considered to evaluate the categorical microaggregation. We also describe the file used and how masked files have been constructed from this file.

The score The score used is the mean of an information loss measure and of a disclosure risk measure. The rationale of this definition is that a good masking method is the one that corresponds to a good trade-off of these both aspects. Such definition is motivated by the fact that information loss and disclosure risk are contradictory properties (no risk usually implies no information, and total information implies total risk).

Disclosure risk was measured (following [5] and [27]) as the number of records re-identified using record linkage programs (we used the mean of the number of re-identifications obtained using two record linkage methods: probabilistic and distance-based). Information loss was also computed as a mean, in this case a mean of several information loss measures. In particular, we considered a direct comparison of categorical values, a comparison of the contingency tables and some entropy-based measures. These measures are described in detail in [5].

Masking methods considered To evaluate the performance of our categorical microaggregation, we have considered 5 alternative masking methods. They are Top and Bottom coding, Rank swapping, Global recoding and the Post-Randomization method (PRAM). For each masking method, we have considered 9 different parameterizations. We briefly describe these methods and the parameterizations considered (see [4] for details).

Top-coding (abbreviated T): This method consists on recoding the last p values of a variable into a new category. We have considered $p = 1, 2, \ldots, 9$.

Bottom-coding (abbr. B): This method is analogous to the previous one but recoding the first p values of a variable into the new category. We have considered $p = 1, 2, \ldots, 9$.

Global recoding (abbr. G): This method recodifies some of the categories into new ones. In our experiment, we have recorded the p categories with lowest frequency into a single one. As before, we have considered $p = 1, 2, \ldots, 9$.

Post-Randomization method or PRAM (abbr. P): Some values are replaced by other values according to a Markov matrix. In our experiments, we have considered the Markov matrix described in [14]. This is, let $T_V = (T_V(1), \ldots, T_V(K))^t$ be the vector of frequencies of the K categories of variable V in the original file (without loss of generality, assume $T_V(K) = \min_k T_V(k)$), let θ be such that $0 < \theta < 1$, then the PRAM matrix for the variable V is defined as:

$$p_{kl} = \begin{cases} 1 - \theta T_V(K)/T_V(k) & \text{if } l = k \\ \theta T_V(K)/((K-1)T_V(k)) & \text{if } l \neq k \end{cases}$$

Let the parameter p be $p := 10\theta$. For each variable we have built nine matrices generated with p taking integer values between 1 and 9.

Rank Swapping (abbr. R): From an operational point of view, this method consists first on ordering the values in ascending order and then replacing each ranked value with another ranked value randomly chosen within a restricted range. For example, the rank of two swapped values cannot differ by more than p percent of the total number of records. We consider values of p from 1 to 9.

The original file and the masked files We have used a data set extracted from the Data Extraction System (DES) of the U. S. Census Bureau [3]. We selected data from the American Housing Survey 1993. Variables and records were selected as follows:

Variables selected: BUILT (Year structure was built), DEGREE (long-term average degree days), GRADE1 (highest school grade), METRO (metropolitan areas), SCH (schools adequate), SHP (shopping facilities adequate), TRAN1 (principal means of transportation to work), WFUEL (fuel used to heat water), WHYMOVE (primary reason for moving), WHYTOH (main reason for choice of house), WHYTON (main reason for choosing this neighborhood). BUILT, DEGREE, GRADE1 were considered ordinal variables and the others nominal.

Records selected: We took the first 1000 records from the corresponding data file. The number of records is small so that repeated experimentation was possible in reasonable time.

For each file, for each masking method and for each parameterization 5 different experiments have been carried out consisting on considering different subsets of variables in the process. This is, we have considered the five subsets of variables described in Table 1. The Table also includes the names we have given to the sets. Note that the set z includes only nominal variables, the set o includes only ordinal variables and the others consider both nominal and ordinal variables.

Variable	*Type*	*g*	*m*	*o*	*p*	*z*
BUILT	*ordinal*			*X*	*X*	
DEGREE	*ordinal*	*X*	*X*	*X*		
GRADE1	*ordinal*			*X*	*X*	
METRO	*nominal*	*X*	*X*			
SCH	*nominal*	*X*	*X*			
SHP	*nominal*	*X*	*X*			
TRAN1	*nominal*	*X*				*X*
WFUEL	*nominal*	*X*				*X*
WHYMOVE	*nominal*				*X*	
WHYTOH	*nominal*	*X*				*X*
WHYTON	*nominal*	*X*				*X*

Table 1. Subsets of variables considered in the experiments

4.2 Experiments

Several parameterizations have been considered for microaggregation. Each parameterization consists on several parameters. Some of the parameters refer to how variables are selected, others control the partition step and some others correspond to the aggregation step. We describe these parameters below divided in these three classes.

Parameters concerning variable selection: a single parameter has been considered for this aspect:

1) **NVar:** corresponds to the number of variables to be aggregated together. This is for multivariate microaggregation (when several variables have to be microaggregated). In this case, groups of NVar variables are considered.

Parameters concerning the partition step: two parameters are used to control our variation of the k-mode algorithm.

1) **K:** is the minimum number of records included in a partition.
2) **NIt:** refers to the maximum number of iterations allowed in the iterative process.

Parameters concerning the aggregation step: four different parameters are used to select the aggregation procedure and to fix it.

1) **Mode?:** in the case of categorical variables on ordinal scales, we can select among the mode and the median aggregation method. When this parameter is set to true, the mode is applied. In the case of nominal scales, only the mode operator is allowed.
2) **Convex?:** this parameter is to permit to make the frequency function convex. When set to true, Equation 1 is applied (instead, when set to false, $acc'(a) = acc(a)$). Recall that making frequencies optional allows compensation among small and larger values because, when using the median, the aggregation of a large and a small value can lead to something in between. This option can only be applied to ordinal variables.
3) **Alpha:** this parameter is used to distort the probabilities using the fuzzy quantifier $Q(x) = x^{\alpha}$. Recall that the use of a fuzzy quantifier allows to increase the importance of large/central or small values. Again, this option can only be applied to ordinal variables.
4) **Random?:** instead of applying the median, a random selection is selected among the categories in the cluster when Random? is set to true. The probability of selecting a particular value is proportional to its frequency. As before, this option can only be applied to ordinal variables.

For each of the parameters above (except for the number of iterations NIt that is fixed to 5) several parameterizations have been considered. In particular, we have considered all aggregation methods and the number of variables (NVar) and the parameter K between 1 and 9. For the parameter α the values 0.2, 0.4, 0.6..., 2.0 have been considered.

The parameterizations considered for the microaggregation together with the experiments considered for all the other masking methods resulted in 24525

Mode?	*Random?*	*Convex?*	*NVar*	*K*	α	*Set*	*Rank*
F	*T*	*T*	04	09	2.0	*p*	1220
T	*T*	*T*	09	09	1.6	*p*	1352
F	*T*	*T*	06	07	0.6	*p*	1455
F	*T*	*T*	06	07	0.4	*p*	1567
F	*T*	*F*	07	07	1.4	*p*	1580
F	*T*	*F*	09	07	0.6	*z*	2696
F	*T*	*F*	04	05	0.2	*z*	2824
F	*T*	*F*	04	05	0.4	*z*	2825
F	*T*	*F*	04	05	0.6	*z*	2826
F	*T*	*F*	04	05	1.0	*z*	2827
F	*T*	*T*	09	09	1.2	*m*	4605
F	*T*	*F*	08	09	0.6	*m*	5483
F	*T*	*F*	09	09	1.2	*m*	5701
F	*T*	*T*	09	09	1.0	*m*	6029
F	*T*	*F*	09	09	1.6	*m*	6087
F	*T*	*F*	08	08	0.6	*o*	1
F	*T*	*F*	09	08	1.0	*o*	2
F	*F*	*F*	04	09	0.6	*o*	3
F	*F*	*F*	09	09	0.6	*o*	4
F	*T*	*T*	04	06	1.6	*o*	5
F	*T*	*T*	09	06	1.2	*g*	14506
F	*T*	*F*	09	07	1.0	*g*	14563
F	*T*	*T*	09	04	1.8	*g*	14585
F	*T*	*T*	09	04	1.0	*g*	14596
F	*T*	*F*	08	05	1.6	*g*	14601

Table 2. Parameterizations that performed the best for microaggregation.

different experiments. The computation of all these experiments lasted 4.5 days (in a PC at 2GHz, running Red Hat 7.0).

Table 2 gives the 5 best parameterizations for the 5 sets of variables. It can be observed that the best results are obtained for Mode?=false (24 times over 1) – this corresponds to the use of the CWOW-median operator, Random?=true (23 times over 2), and Convex=false (15 times over 10) and a large number of variables (9 variables is the most selected NVar), and K=9. The most frequent value for the α parameter is 0.6 (being 0.8 the second one).

This table also gives (column Rank) the position in a global ranking considering (method, parameterization, original file). Table 3 gives the two best parameterizations for all the other masking methods tested (B, T, G, R, P), using each of the selected set of variables. In this table, P(Set) refers to the best parameterization obtained for the masking method with the set of variables Set, and Rank(Set) refers to the position in the ranking of such parameterization for the same set. It can be observed that the best parameterizations in Table 2 perform better than the parameterizations of the other methods except for the set g.

According to all this, a categorical microaggregation with parameters Mode? equal to false and Random? equal to true yields good results. A good set of parameters is when the number of variables is large (*e.g.*, NVar=9) and the constants K and α are such as: $K = 9$ and $\alpha = 0.6$.

In the case of the set g, the best performance corresponds to the PRAM masking method. Note that the set g precisely corresponds to the set with a major number of variables.

Using Table 3, we can see that Rank Swapping can be considered as the second best masking method. We observe that when the set of variables considered corresponds to p, m or o, rank swapping has the second best performance. These three sets of variables correspond to the case in which all or most of the variables are ordinal. The two other sets analyzed z and g, with bad results, correspond to either all variables nominal or one ordinal over 7 nominal ones.

$Method$	$P(p)$	$Rank(p)$	$P(z)$	$Rank(z)$	$P(m)$	$Rank(m)$	$P(o)$	$Rank(o)$	$P(g)$	$Rank(g)$
B	9	21548	1	16534	4	23682	7	12996	1	23719
B	8	22158	7	18828	3	23761	9	13183	2	24503
T	9	17318	7	7075	3	21936	6	13504	1	11660
T	8	19744	6	7319	2	22714	5	14367	2	12351
G	9	21703	6	7277	4	22950	7	11872	2	11233
G	8	22452	5	7656	3	23400	8	12199	1	11863
R	6	9567	1	9664	3	11423	2	9797	2	19998
R	9	10779	7	10917	7	12000	3	11139	1	20109
P	4	23590	9	9161	9	20945	9	21789	9	10473
P	6	23604	5	9189	8	21605	8	22187	6	11079

Table 3. For each masking method considered, except for microaggregation, the best two parameterizations in terms of the score

5 Conclusions and future work

The results presented here expand the ones in [5]. In this work, two additional masking methods (namely, Rank Swapping and Categorical microaggregation) have been added. In that paper, it was concluded that the PRAM performance (with the current parameterization) was not good. In this work we have shown that for a particular set of variables PRAM yields the best results. These results have been obtained for the largest number of variables. Nevertheless, further work is needed to confirm the influence of a large number of variables on the good performance of PRAM.

Categorical microaggregation has been shown to have a good performance. This method is based on the k-modes clustering algorithm and either the mode or the median for the aggregation step. For each of the sets of variables except one, there was a good parameterization that yielded to the best performance.

Additionally, the analysis described in this paper show that rank swapping is the second best masking method for ordinal data. This method was not included in the analysis in [5].

In this work we have studied masking methods using general information loss and disclosure risk measures. This is, not considering particular data uses. The analysis of microaggregation from this viewpoint remains as future work.

As future work we consider the application of fuzzy clustering algorithms in the clustering partition step. Recent results on fuzzy clustering are reported in *e.g.* [2, 8, 13, 15].

Acknowledgements

Work partly funded by the European Union (project "CASC" IST-2000-25069) and the MCYT (project TIC2001-4051-E).

References

1. Agrawal, R., Srikant, R., (2000), Privacy preserving data mining, Proc. 2000 ACM SIGMOD Int'l Conf. Management of Data, ACM Press, 439–450.
2. J.-H. Chiang, P.-Y. Hao, (2003), A new kernel-based fuzzy clustering approach: support vector clustering with cell growing, IEEE Trans. on Fuzzy Systems, 11:4 518- 527
3. Data Extraction System (DES), U. S. Census Bureau, http://www.census.gov/DES/www/welcome.html,
4. Domingo-Ferrer, J., Torra, V., (2001), Disclosure Control Methods and Information Loss for Microdata, 91-110, in Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies, P. Doyle, J. I. Lane, J. J. M. Theeuwes, L. M. Zayatz (Eds.), Elsevier.
5. Domingo-Ferrer, J., Torra, V., (2001), A Quantitative Comparison of Disclosure Control Methods for Microdata, 111-133, in Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies, P. Doyle, J. I. Lane, J. J. M. Theeuwes, L. M. Zayatz (Eds.), Elsevier.
6. Domingo-Ferrer, J., Torra, V., (2003), Median based aggregation operators for prototype construction in ordinal scales, Intl. J. of Intel. Syst., 18:6 633-655.
7. Doyle, P., Lane, J. I., Theeuwes, J. J. M., Zayatz, L. M. (Eds.), (2001), Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies, Elsevier.
8. Eschrich, S., Ke, J., Hall, L.O., Goldgof, D.B., (2003), Fast accurate fuzzy clustering through data reduction, IEEE Trans. on Fuzzy Systems, 11:2 262- 270.
9. Felso, F., Theeuwes, J., Wagner, G. G., (2001), Disclosure Limitation Methods in Use: Results of a Survey, 17-42, in Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies, P. Doyle, J. I. Lane, J. J. M. Theeuwes, L. M. Zayatz (Eds.), Elsevier.
10. Godo, L., Torra, V., (2000), On aggregation operators for ordinal qualitative information, IEEE Trans. on Fuzzy Systems, 8:2 143-154.
11. Herrera, F., Herrera-Viedma, E., Verdegay, J.L., (1995), A Sequential Selection process in Group Decision Making with a Linguistic Assessment Approach, Information Science, 85 (1995) 223-239.

12. Huang, Z., Ng, M. K., (1999), A fuzzy k-modes algorithm for clustering categorical data, IEEE Trans. on Fuzzy Systems, 7:4 446-452.
13. Kolen, J.F., Hutcheson, T., (2002), Reducing the time complexity of the fuzzy c-means algorithm, IEEE Trans. on Fuzzy Systems 10:2 263-267.
14. Kooiman, P., Willenborg, L., Gouweleeuw, J., (1998), PRAM: a method for disclosure limitation of microdata, Statistics Netherlands, Research Report.
15. Leski, J.M., (2003), Generalized weighted conditional fuzzy clustering, IEEE Trans. on Fuzzy Systems, 11:6 709-715.
16. Miyamoto, S., (1999), Introduction to fuzzy clustering, Ed. Morikita, Japan.
17. Miyamoto, S., Umayahara, K., (2000), Methods in Hard and Fuzzy Clustering, pp 85–129 in Z.-Q. Liu, S. Miyamoto (Eds.), Soft Computing and Human-Centered Machines, Springer-Tokyo.
18. Sande, G., (2002), Exact and approximate methods for data directed microaggregation in one or more dimensions, Int. J. of Unc. Fuzziness and Knowledge Based Systems, 10:5 459-476.
19. Sugeno, M., (1974), Theory of Fuzzy Integrals and its Applications. (PhD Dissertation). Tokyo Institute of Technology, Tokyo, Japan.
20. Torra, V., (1996), Negation functions based semantics for ordered linguistic labels, Intl. J. of Intel. Syst., 11 975-988.
21. Torra, V., (1997), The Weighted OWA operator, Intl. J. of Intel. Syst., 12 153-166.
22. Torra, V., (2001), Aggregation of linguistic labels when semantics is based on antonyms, Intl. J. of Intel. Systems, 16 513-524.
23. Willenborg, L., De Waal, T., (1996), Statistical Disclosure Control in Practice, Springer LNS 111.
24. Willenborg, L., de Waal, T., (2001), Elements of Statistical Disclosure Control, Lecture Notes in Statistics, Springer-Verlag.
25. Winkler, W. E., (1995), Matching and record linkage, *Business Survey Methods* (ed. B. G. Cox), New York: Wiley, 355-384.
26. Xu, Z. S., Da, Q. L., (2003), An overview of operators for aggregating information, Int. J. of Intel. Systems 18 953-969.
27. Yancey, W. E., Winkler, W. E., R. H. Creecy, (2002), Disclosure Risk Assessment in Perturbative Microdata Protection, Lecture Notes in Computer Science, 2316, 135-152.

Median based aggregation operators for prototype construction in ordinal scales

Josep Domingo-Ferrer
Dept. Computing Engineering and Maths - ETSE
Universitat Rovira i Virgili
Av Països Catalans 26
E-43007 Tarragona, Catalonia
jdomingo@etse.urv.es

Vicenç Torra
Institut d'Investigació en Intel·ligència Artificial - CSIC
Campus Universitat Autònoma de Barcelona
E-08193 Bellaterra, Catalonia
vtorra@iiia.csic.es

April 9, 2002

Abstract

This paper studies aggregation operators in ordinal scales for their application to clustering (more specifically, to microaggregation for statistical disclosure risk). In particular, we consider these operators in the process of prototype construction. The paper analyses main aggregation operators for ordinal scales (plurality rule, medians, Sugeno integrals and ordinal weighted means among others) and shows the difficulties for their application in this particular setting. Then we propose two approaches to solve the drawbacks and we study their properties. Special emphasis is given to the study of monotonicity as the operator is proven to non satisfy this property. Exhaustive empirical work shows that in most practical situations this cannot be considered a problem.

Keywords: aggregation operators, median operators, OWA-like operators, WOWA-like operators, monotonicity, clustering, microaggregation, ordinal scales

1 Introduction

Information fusion techniques and aggregation operators are commonly applied into several fields of human knowledge. As different fields imply different requirements, a large number of aggregation operators exists nowadays. Also, differences on the way knowledge is represented forced to the development of tools to deal with the different knowledge representation formalisms. In particular, methods exist to deal with different kind of data. For example, there are methods to fuse numerical information (i.e., data in numerical scale [9]), categorical information (either ordinal [10] or nominal scales [15]), information expressed by means of partitions (or, equivalently, equivalent relations [6]), dendrograms (classification trees), preferences, orderings, images, ...

This work is devoted to the case of categorical information. The development of operators of any kind for categorical information is always a difficult task due to the limited number of commonly established operators over these scales. In the particular case of aggregation operators, this is even more noticeable because the corresponding operators over numerical scales are the means. These well-known operators are based on product and addition, two operations that do not apply to ordinal scales.

To overcome these difficulties, researchers have considered three main different approaches for the case of ordinal scales. We detail them below considering operators over the scale $L = \{l_0, \cdots, l_R\}$ where $l_0 \leq_L l_1 \leq_L \cdots \leq_L l_R$. This classification is based on [19].

1. *Explicit quantitative or fuzzy scales:* It is assumed a translation function that assigns values in a different numerical scale for all values in the original ordinal scale. The operators in the ordinal scale are defined from the operators in the underlying scale. Operators defined for fuzzy sets using the extensional principle belong to this class. In some cases, this explicit scale is not given but inferred from additional knowledge about the ordinal scale (e.g. one-to-many negation functions [17]). This is the case of the aggregation operator in [19].

2. *Implicit numerical scale:* Operators assume an implicit numerical scale underlying the ordinal scale where values are defined. Usually, each category l_i is dealt as the corresponding integer i. This is the case of Linguistic OWA [11] and Linguistic WOWA [18].

3. *Operating directly on qualitative scales:* Operators stick to a purely ordinal scale and are based only on operators of this scale. This is the case of the median operator or the Sugeno integral [16]. These operators only use the relation $\leq_L$ and *minimum* and *maximum* that rely on $\leq_L$. Other operators in this class (e.g., the weighted mean defined in [7]) are based on t-norms and t-conorms. Two operators that can be defined axiomatically over ordinal scales.

The motivation of our work is the application of aggregation operators to statistical disclosure risk. In particular, we consider the extension of existing microaggregation procedures for numerical scales to ordinal scales (see [4] for a state of the art description of microaggregation procedures). Microaggregation techniques are applied to avoid disclosure of confidential data. To avoid the re-identification of the individual in a data file, the information of these individuals is distorted. Microaggregation consists on clustering the data in smalls clusters (less than 10 individuals) and replacing the original values by the prototype (an aggregated value) of the cluster. See [3] for a detailed analysis of the performance of microaggregation with respect to other distorting techniques for microdata protection.

In this setting, tipically, no much information is available on the underlying semantics of categories in ordinal scales. This focus our work on the third class of aggregation operators. This is the only case where no assumptions are made on the existence of an underlying structure beneath the ordinal scale.

The structure of this work is as follows. We begin reviewing in Section 2 existing aggregation operators in ordinal scales. This section also reviews different usages of weights in aggregation operators. Then, in Section 3, we comment on the suitability of these operators for prototype building. Section 4 introduces new operators for solving the shortcommings of existing ones, and analyzes their properties. The work finishes with some conclusions.

2 Aggregation operators in ordinal scales

In this section, we review some of the existing aggregation operators in ordinal scales that operate directly on categorical values. We begin with the plurality rule. Then we follow with the median and the Sugeno integral. The Sugeno integral generalizes the median and other aggregation operators in categorical scales. We finish outlining the ordinal weighted mean.

2.1 Plurality rule

The Plurality rule (or plurality function) corresponds to the selection of the most frequent elements. In fact, the definition does not return a single element but the set of elements that appear more often. Assuming that values to be aggregated belong to the set L, the plurality rule can be formulated in the following terms (this definition is based on [15]):

Definition 1 *A mapping $P : L^N \to \wp(L)$ is a* plurality function *when $P(a_1, \cdots, a_N)$ is the set of all those y in L so that no z in L appears more often in $(a_1, \cdots, a_N)$ than y.*

This definition shows that the procedure can be applied to elements in ordinal scales as well as to elements in nominal scales. So, L is not required to be ordered.

Plurality rule can be extended to introduce weights to measure the reliability of or the confidence in each value a_i. This is formulated making explicit the information sources $X = \{x_1, \cdots, x_N\}$ (here we assume that x_i supplies the value a_i) and defining the weights as either a function $\mathbf{w}$ from X into a given domain (e.g., $[0, 1]$) or as a weighting vector $\mathbf{w} = (w_1, \cdots, w_N)$. Both approaches are equivalent as $w_i = w(x_i)$. In the definition of the weighted plurality function it is also considered a function f to relate each information source with the value it supplies: $f(x_i) = a_i$.

With all this information, the weighted plurality rule selects the values that accumulate more weights. This is formalized below by means of a function *acc* that when applied to $a \in L$ returns the accumulation of the weights of all the sources x_j that supply the value a.

Definition 2 *Let* $\mathbf{w}$ *be a weighting vector of dimension* N*, then a mapping* $WP_{\mathbf{w}} : L^N \to \wp(L)$ *is a* weighted plurality function *when* $P_{\mathbf{w}}(a_1, \cdots, a_N)$ *is the set of all those* y *in* L *so that no* z *in* L*,* $acc(z) > acc(y)$ *where* $acc(a) = \sum_{f(x_j)=a} w(x_j)$

In this definition, the range of the weights is restricted to be in such a way that addition is allowed. Therefore, real numbers and integer numbers are both appropriate for weighting vectors. Moreover, ordinal scales where addition-like operators are defined are also appropriate. This is the case of ordinal scales with t-conorms (see [14] for a detailed analysis of t-norms and t-conorms in ordinal scales). We would like to underline that there is no need to impose that the domain of the weights are equal to the one of the data.

2.2 Median

The median procedure is to select the element that occupies the central position of a sequence of elements when they are ordered according to their value. This can be formaly described for numerical data as follows:

Definition 3 *A mapping M:* $\mathbb{R}^N \to \mathbb{R}$ *is a* median *of dimension* N *if:*

$$M(a_1, \cdots, a_N) = \begin{cases} \frac{a_{\sigma(N/2)} + a_{\sigma(N/2+1)}}{2} & \textit{when } N \textit{is even} \\ a_{\sigma(\frac{N+1}{2})} & \textit{when } N \textit{is odd} \end{cases}$$

where $\{\sigma(1), ..., \sigma(N)\}$ *is a permutation of* $\{1, ..., N\}$ *such that* $a_{\sigma(i-1)} \geq a_{\sigma(i)}$ *for all* $i = \{2, ..., N\}$ *(i.e.* $a_{\sigma(i)}$ *is the i-th largest element in the collection* $a_1, ..., a_N$*).*

When dealing with categorical data (this is, M is a function $M : L^N \to L$), one of the following expressions will be used for the case of N being even:

$$a_{\sigma(\lfloor \frac{N+1}{2} \rfloor)} \qquad a_{\sigma(\lceil \frac{N+1}{2} \rceil)}$$

They correspond, respectively, to $a_{\sigma(N/2)}$ and to $a_{\sigma(N/2+1)}$.

This definition can also be extended to include weighting vectors. In this case, the central element is a relative position according to the weights. As in the case of the Plurality rule, we formalize this definition considering the set of sources X, the function f that assigns the values to the sources and the weighting vector $\mathbf{w}$.

Definition 4 *Let* $\mathbf{w}$ *be a weighting vector of dimension* N*, then a mapping* $WM_{\mathbf{w}} : \mathbb{L}^N \to \mathbb{L}$ *is a* Weighted Median *of dimension* N *if:*

$$WM_{\mathbf{w}}(a_1, \cdots, a_N) = a \text{ if and only if } acc(a) > 0.5 \geq acc(b)$$

where acc is a function over the values in $\{a_1, ..., a_N\}$ *defined as* $acc(a) = \sum_{f(x_j) \leq a} w(x_j)$ *and where b is the largest element in* $\{a_1, ..., a_N\}$ *that is smaller than a. This is,* $b = \max\{x | x \in \{a_1, ..., a_N\}, x < a\}$.

In this case, the most natural weighting vector is one defined by positive real numbers that add to one. This is, $\sum w_i = 1$ and $w_i \in (0, 1]$ (note that the definition requires $w_i \neq 0$). However, other possibilities are also possible. In particular, natural numbers can be considered. The weighted median for weights in $\mathbb{N}$ can easily be translated into the previous one through normalization. This is, defining a new weighting vector $w_i' = w_i / \sum_j w_j$. Moreover, an ordinal scale O with multi-valued logic operators can also be used. In this case, besides of a t-conorm for addition, an involutive negation is also required (a function n from O to O). In such case, instead of selecting a value on the basis of the value 0.5 we would use the element $x \in O$ such that its negation is also x (i.e., $x = n(x)$).

2.2.1 Order statistics

There exists a set of aggregation operators that are similar to the median. They are the so-called order statistics (we denote this family of functions by OS). Order statistics permit the selection of the i-th greatest value. To do so, the operator requires a preliminary ordering process as in the case of the median and then an integer value i in the range $[1, N]$ to select the i-th element. Alternatively, a definition can be given when instead of an integer value, a real number α in the unit interval is given. I.e., selecting the element that occupies the $\alpha \cdot 100$ percentage of the domain. As the operator only relies on the ordering, it can be applied to ordinal scales.

When the selection of an element is based on a real number (in the unit interval) weights can be included in the definition. This corresponds to replace 0.5 by α in Definition 4. We denote by WOS the corresponding weighted order statistics. It is clear that the approach is similar to the case of the weights in the median. As before, weights correspond to the importance of the sources and can either be real or natural numbers. In the latter case, normalization is required. Ordinal scales can also be used. In this case, the parameter i should be a value in the same ordinal scale (instead of a real number in the unit interval).

2.3 Sugeno integral

An alternative aggregation operator that also permits the inclusion of weights for the information sources is the Sugeno integral [16] (see [13] for a detailed account of its properties). However, this integral does not consider weighting vectors but the so-called fuzzy measures. If $X = \{x_1, \cdots, x_N\}$ is the set of information sources, a fuzzy measure is a set function that given a subset A of X returns a measure of its importance.

Fuzzy measures satisfy three axioms: (i) the measure of the empty set is zero (when no source is considered, the importance is zero), (ii) the measure of the whole set is 1 (when all the sources are considered, the importance is maximal and settled to one); and (iii) the larger the set of sources, the larger its importance. The first two conditions correspond to boundary conditions and the third one corresponds to monotonicity. Formal definition of these conditions are given below:

Definition 5 *A fuzzy measure μ on a set X is a set function $\mu : \wp(X) \to [0,1]$ satisfying the following axioms:*

(i) $\mu(\emptyset) = 0$, $\mu(X) = 1$ (boundary conditions)

(ii) $A \subseteq B$ implies $\mu(A) \leq \mu(B)$ (monotonicity)

This definition is given in the interval $[0,1]$, but the same definition applies to any ordinal scale $L = \{l_0, \cdots, l_R\}$. In this latter case, the measure is a function from $\wp(X)$ into L and the boundary conditions are $\mu(\emptyset) = l_0$ and $\mu(X) = l_R$.

The Sugeno integral [16] is defined as the integral of a function f (the one that establishes the value $f(x_i)$ for the information source x_i) with respect to a fuzzy measure. In a numerical scale, the definition is as follows:

Definition 6 *Let μ be a fuzzy measure on X, then, the* Sugeno integral *(SI for short) of a function $f : X \to [0,1]$ with respect to μ is defined by:*

$$(S) \int f d\mu = \max_{i=1,N} \min(f(x_{s(i)}), \mu(A_{s(i)})) \tag{1}$$

where $f(x_{s(i)})$ indicates that the indices have been permuted so that $0 \leq f(x_{s(1)}) \leq ... \leq f(x_{s(N)}) \leq 1$, $A_{s(i)} = \{x_{s(i)}, ..., x_{s(N)}\}$ and $f(x_{s(0)}) = 0$.

When the values belong to an ordinal scale, an analogous definition applied. In this latter case it is important to emphasize that both the function f and the fuzzy measure μ are defined as mappings into the same ordinal scale L otherwise the minimum and the maximum operators are not meaningful.

The Sugeno integral is a very general operator as it generalizes several other aggregation operators. In particular, it generalizes the weighted minimum and the weighted maximum (see [5] for a detailed description of these operators and of their properties). They are aggregation operators to be used to model logical conjunction and disjunction when the sources are weighted. We review below the weighted maximum. The weighted minimum has a similar definition. Both operators use weighting vectors for expressing importance or reliability. Here the weights map each source into a value in an ordinal scale. Note that, as before, the scale for the values to be aggregated should be the same that the scale for the weights. This is so because the minimum combines the values of the weighting vector and the values a_i.

Definition 7 *A vector* $v = (v_1 ... v_N)$ *is a* possibilistic weighting vector *of dimension* N *if and only if* $v_i \in L$ *and* $\max_i v_i = l_R$*.*

Definition 8 *Let* $\mathbf{u}$ *be a weighting vector of dimension* N*, then a mapping WMax:* $L^N \rightarrow L$ *is a* weighted maximum *of dimension* N *if* $WMax_{\mathbf{u}}(a_1, ..., a_N) = \max_i \min(u_i, a_i)$*.*

2.4 Ordinal weighted mean

In this section, we give an overview of ordinal weighted mean without going into details. See [7] for detailed definitions and properties and [8] for an extension of the approach to Choquet integrals.

The ordinal weighted mean (OWM for short) is a different approach to extend the weighted mean to ordinal scales. The general idea of the operator is to translate addition and product in the weighted mean by similar operations in the ordinal scale. Two operations of multi-valued logics are selected for this purpose: t-norms and t-conorms.

T-conorms are addition-like operators that satisfy monotonicity, commutativity, associativity and have as neutral element the value 0 (l_0 in the ordinal scale $L = \{l_0, \cdots, l_R\}$). T-norms are product-like operators that satisfy the same properties except for the neutral element that in this case is 1 (l_R when defined in the ordinal scales L).

Ordinal weighted mean assumes that weights are natural numbers. Then, the multiplication of a weight by a value corresponds to multiple additions of the corresponding value. Here addition is achieved through the t-conorm. As the ordinal scale is usually not enough to accumulate all the values to be aggregated, a new scale is introduced that extends the original scale. This new scale is the product of the subset of natural numbers $\{1, \cdots, N\}$ (where N is the number of values to be aggregated) and the original scale. Once the accumulated value is obtained in this new scale, division by the accumulation of the weights leads to the final aggregated value.

Extensions of this operator exist that consider other scales than natural numbers for the weights. Also, the same approach was applied to extend the Choquet integral [2] to ordinal scales. This is the so-called Ordinal Choquet integral (OCI for short). Choquet integral is the natural extension of the weighted mean to the case of considering numerical fuzzy measures. In some way, Sugeno integrals are the ordinal counterpart of Choquet integrals.

2.5 Considering weights in aggregation operators

Aggregation operators use parameters for expressing additional knowledge about the values, the sources and its current application. Some of the common uses of the parameters are the following ones:

Expressing importances of individual information sources: This is the typical case of weighting vectors in weighted means and similar aggregation operators (weighted maximum, weighted minimum, plurality rule, median). We associate to each source a weight in a given scale. The larger the weight, the more important is the source in determining the aggregated value.

Expressing importances of values: This is the approach considered in the OWA operator (operator defined by Yager in [20] – see also [21] about including other types of weights). Weights do not measure the importance of a source but of the values. For instance, it is possible to give more importance to small values than to larger ones. This would be the case if a robot fuses estimated distance to a nearby object: it is more important to consider small values than larger ones to avoid collisions. OWA operators and related ones (e.g., Choquet integral that generalizes OWA operators) can be used for this purpose.

Expressing importances of sets of information sources: This is the case of the Sugeno integral and other similar operators (the Choquet integral and the Fuzzy t-integral). These operators do not only allow to express the importance of a particular information source, but also the importance of a set of sources. Fuzzy measures can be used to represent this information. In the numerical case, it can be proven that fuzzy measures can be used to represent both the importances of the individuals and the importance of the values.

	P	WP	M	WM	OS	WOS	SI	$WMax$	OWM	OCI
(1)	$\surd$	w	$\surd$	w	if $i \in I$	w, if $i \in I$	μ	π	w $(O, \oplus, \otimes, n)$	μ $(O, \oplus, \otimes, n)$
(2)	0	N	0	N	1	$N+1$	2^N	N	N	2^N
(3)		$\mathbb{R}, \mathbb{N}$ $(O, \oplus)$		$\mathbb{R}, \mathbb{N}$ $(O, \oplus, n)$		$\mathbb{R}, \mathbb{N}$ $(O, \oplus, n)$	L	L	$\mathbb{N}$ $(O, \oplus)$	$\mathbb{N}$ $(O, \oplus)$
(4)	X	$\surd$	X	$\surd$	X	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
(5)	X	X	X	X	X	X	$\surd$	X	X	$\surd$
(6)	X	X	X	X	X	X	X	X	$\surd$	$\surd$

Table 1: Characteristics of ordinal aggregation operators: $\surd$ means that the characteristic is always fullfilled; X that is never possible; other values correspond to particular characteristics. Here, I stands for the unit interval, O corresponds to an arbitrary ordinal scale, $(O, \oplus)$ to an ordinal scale with a t-conorm, $(O, \oplus, n)$ an arbitrary ordinal scale with a t-conorm and a negation and $(O, \oplus, \otimes, n)$ an ordinal scale with a t-conorm, a t-norm and a negation.

2.6 Summary of aggregation operators in ordinal scales

Table 2.5 gives an overview of the main characteristics of the aggregation operators reviewed so far.

The first row is whether the function can be used for an arbitrary number of values to be aggregated and the parameters required, if any. In fact, all functions can be applied to an arbitrary number of parameters easily. In the case of the order statistics, it is appropriate that the parameter used is a real number in the unit interval in order that the selection of the i-th element do not change the meaning when additional elements are considered. With a real number, the parameter corresponds to the selection of the element that occupies the $i\%$ percent.

The second row is the number of parameters required when the number of values to be aggregated is N.

The third row is the range of the weights (if any). In this row, O corresponds to an arbitrary ordinal scale while L is used when the scale should be the same that the one for the values to be aggregated. $\oplus$, $\otimes$ and n stand for t-conorm, t-norm and negation functions over O. $\mathbb{R}$ and $\mathbb{N}$ stand, as usual for real and natural numbers.

The fourth row is whether the aggregation procedure allows the weighting of the sources. The fifth row is for the weighting of the values. Positive marks are given for the Sugeno and the Choquet integral to both kind of weights as fuzzy measures can be defined to express this information. However, for measures in ordinal scales it is difficult to model at the same time the weighting of the sources and the weighting of the elements. This is not the case in the numerical setting when the measure can be built from two weighting vectors one modeling each alternative (as for the WOWA in [18]).

The last row is about the possibility of obtaining a value that is not present in the original set of values to be aggregated.

3 Aggregation procedures for prototype construction

In this Section, we review the difficulties of using the aggregation procedures reviewed so far when applied to building prototypes within clustering methods. Although our point of view is biased to clustering methods for microdata protection, the analysis is appliable to most clustering problems.

Clustering methods are applied to multidimensional data to build a set of clusters in which similar elements are put together and dissimilar elements are left into different classes. One of the open problems in clustering is how to deal with categorical data. In fact, several difficulties arise in this case: computation of similarities between categories, combination of similarities when each individual is represented in terms of different variables evaluated in different scales, prototype calculation for each cluster. In this work we are interested in the lattest problem: the computation of the cluster prototype.

The computation of the prototype is usually achieved in numerical scales using some kind of aggregation procedure. Usually an arithmetic mean although some other aggregation operators are conceivable. In particular, the weighted mean (e.g. to give different importance to different individuals in the cluster [12]) or the OWA (e.g. to give more importance to central elements than to elements with large or small values [22]).

In the case of categorical data, the methods described in Section 2 are applicable. Now we consider in detail the applicability of each method for prototype calculation:

Plurality rule: The application of the majority rule is straightforward. However, some inconveniences can be distinguished. The first one is that the majority rule returns a set of the most frequent values. Therefore, when the prototype is a single value, a selection procedure has to be considered to select one of the values. Another drawback is that the function does not allow for compensation. We understand here for compensation the fact that when the data to be fused contains two values a_i and a_j, the output can be a value in between, say a_k, regardless $a_k \in \{a_1, \cdots, a_N\}$ or not. In other words, an aggregation function $\mathbb{C}$ is not compensative if for all $a_i, a_j \in \{a_1, \cdots, a_N\}$, the aggregated value is always one of the original ones: $\mathbb{C}(a_1, \cdots, a_N) \in \{a_1, \cdots, a_N\}$ for all $a_i \in L$

Therefore, when large and small values but not medium ones are fused, the final value will be either a large or a small one. Note that in the numerical case, the mean $\bar{x} = \sum_i x_i/N$ minimizes $\sum_i (x_i - \bar{x})^2$, and the selection of a large value (or a small one) instead of $\bar{x}$ would give a larger difference.

An additional difficulty of this lack of compensation is that when the number of values to aggregate is small, small variations on the elements can provoke large modifications of the output. E.g., the aggregation of the values l_0, l_0, l_3, l_4 is l_0 and the aggregation of the values l_4, l_0, l_3, l_4 is l_4. Thus, a small modification of the inputs (a single value) results into a large variation of the output (from l_0 to l_4).

Weighted median has an additional difficulty: it is not always possible to have available the required weighting vector. This is so, because in prototype selection it would be required a weight for each individual. In fact, there are some applications in which this information is available (e.g., [12]). However, this is not the general case, because it is usually assumed that the representativeness of all elements is the same (for all the application domain).

Special difficulties arise when weights are not numerical but defined in ordinal scales. This is so, because not all the clusters have the same number of elements and therefore, normalization is required in each cluster (otherwise with a few elements we can get that all elements a_i have $acc(a_i)$ equal to 1, and, therefore, selection is not possible). Also, selection of the appropriate t-conorm is not an easy task, specially for non-experienced users.

Also related to weights, no weights for the values are considered in the function.

Median: The application of the median operator for prototype selection is straightforward. However, it presents some of the drawbacks of the plurality rule: The median always returns one of the values to be aggregated (e.g., the median of l_0, l_{N-1}, l_N is l_{N-1} while a straight average of the indices gives $l_{(2M-1)/3}$); it does not allow to consider weights for the values; and the same comments about the weighting vector given for the plurality rule apply to this case. Order statistics have similar properties although in this case, the weight allows the selection of other values than the central one.

Sugeno integral: The main difficulty for the application of the Sugeno integral in the setting of prototype selection is the definition of the corresponding fuzzy measure. According to the definition of the integral, the fuzzy measure has to be defined into L as the values a_i are. Several difficulties apply in this case: defining measures for all possible clusters requires a huge number of fuzzy measures (only parameterized families of fuzzy measures can be used - and parameterization is difficult in ordinal scales); when several variables are used in the clustering process, fuzzy measures have to be defined for each variable (the set L usually changes for each variable and the fuzzy measure has to be defined on the same scale that the variable) and this increases the complexity of this definition; for each variable and each set of N sources, 2^N values are required.

Another drawback of the Sugeno integral is that it does not allow for compensation. In fact, this statement has to be tinged because the final value can be different from the original ones. This is possible because the final value can be one of the ones used by the fuzzy measure. This can cause some sort of compensation.

Some of these difficulties also apply to Weighted maximum.

Ordinal weighted mean: The main difficulty for using the ordinal weighted mean is the requirement of a t-norm and t-conorm for the domains of the variables. This means having one pair (t-norm,t-conorm) for each of the variables.

As a conclusion, we can say that the two most relevant difficulties for applying the above mentioned aggregation operators is that most operators do not allow for compensation and that also most of them do not allow for weighting the sources.

Detailed analysis of the methods shows that the most relevant operation for the problem of prototype selection is the median. This is, in fact, the operator usually considered as the ordinal counterpart of the weighted mean. Sugeno integral and ordinal weighted mean are specially difficult to apply due, respectively, to the need of fuzzy measures and definitions of t-norms and t-conorms.

In the next section we introduce WOW-operators for including compensation and weighting for the sources to categorical aggregation operators. Then we particularize the approach to the case of the median.

4 Weighting of values and compensation

The inclusion of the weights for the values is based on the Weighted OWA (WOWA) operator defined in [17]. This operator is a generalization of both the weighted mean (WM) and the Ordered Weighted Averaging (OWA) operator (defined by Yager in [20]) allowing users to have in a single operator the parameters of both operators. In fact, both WM and OWA have parameters of the same form (weighting vectors: positive weights that add to one). However, in spite of having the same form, the parameters have different meaning. Let us recall both operators:

Definition 9 *A vector $v = (v_1 ... v_N)$ is a* weighting vector *of dimension N if and only if $v_i \in [0,1]$ and $\sum_i v_i = 1$.*

Definition 10 *Let $\mathbf{p}$ be a weighting vector of dimension N, then a mapping WM: $\mathbb{R}^N \rightarrow \mathbb{R}$ is a* weighted mean *of dimension N if $WM_{\mathbf{p}}(a_1, ..., a_N) = \sum_i p_i a_i$.*

Definition 11 *Let $\mathbf{w}$ be a weighting vector of dimension N, then a mapping OWA: $\mathbb{R}^N \rightarrow \mathbb{R}$ is an* Ordered Weighting Averaging (OWA) operator *of dimension N if*

$$OWA_{\mathbf{w}}(a_1, ..., a_N) = \sum_{i=1}^{N} w_i a_{\sigma(i)}$$

where $\{\sigma(1), ..., \sigma(N)\}$ is a permutation of $\{1, ..., N\}$ such that $a_{\sigma(i-1)} \geq a_{\sigma(i)}$ for all $i = \{2, ..., N\}$ (i.e. $a_{\sigma(i)}$ is the i-th largest element in the collection $a_1, ..., a_N$).

Similarities and differences between both operators can be underlined as follows:

- The weighted mean is a linear combination of weights and values where the weights are linked to the values we aggregate. This is usually understood as the importance or reliability of the information sources. The larger a weight is, the more influence has the corresponding value to the final output. The smaller a weight, the lesser influence has the correponding value.
- The OWA operator is also a linear combination of weights and values. However, in this operator weights are not linked to the values themselves but on their relative position. Note that any permutation π of the values to be aggregated lead to the same result: $OWA_{\mathbf{p}}(a_1, ..., a_N) = OWA_{\mathbf{p}}(a_{\pi(1)}, ..., a_{\pi(N)})$.

The WOWA operator that generalizes both operators is defined as follows:

Definition 12 *Let* $\mathbf{p}$ *and* $\mathbf{w}$ *be two weighting vectors of dimension* N*, then a mapping WOWA:* $\mathbb{R}^N \rightarrow \mathbb{R}$ *is a* Weighted Ordered Weighted Averaging (WOWA) operator *of dimension* N *if*

$$WOWA_{\mathbf{p},\mathbf{w}}(a_1, ..., a_N) = \sum_i \omega_i a_{\sigma(i)}$$

where σ *is defined as in the case of the OWA (i.e.,* $a_{\sigma(i)}$ *is the i-th largest element in the collection* $a_1, ..., a_N$*), and the weight* ω_i *is defined as:*

$$\omega_i = w^*(\sum_{j \leq i} p_{\sigma(j)}) - w^*(\sum_{j<i} p_{\sigma(j)})$$

with w^* *being a monotonic increasing function that interpolates the points* $(i/N, \sum_{j \leq i} w_j)$ *together with the point* $(0,0)$*. The function* w^* *is required to be a straight line when the points can be interpolated in this way.*

In this definition, the weighting vector $\mathbf{p}$ corresponds to the weighting vector of the weighted mean and $\mathbf{w}$ corresponds to the weighting vector of the OWA operator. Then, ω is a new weighting vector that considers the interactions between $\mathbf{p}$ and $\mathbf{w}$.

The function w^* built above from the vector $\mathbf{w}$ can be understood as a fuzzy quantifier (a non-decreasing fuzzy quantifier) while the weights $\mathbf{p}$ can be seen as a probability distribution. A non-decreasing fuzzy quantifier is a monotonic function Q (i.e., $Q(a) \geq Q(b)$ for all $a > b$) such that $Q(0) = 0$ and $Q(1) = 1$.

In the definitions given above, weighting vectors are presented in conjunction with the definition of the operator. However, these vectors and their transformation can be established without the corresponding operator and used in other families of operators. This is defined below using the non-decreasing fuzzy quantifier Q (Q can be interpolated from $\mathbf{w}$ when required as above).

Definition 13 *Let* $(a_i, p_i)_{i=1,N}$ *be a pair defined by a value and the importance of* a_i *expressed in a given domain* $D \subset \mathbb{R}^+$*, and let* Q *be a fuzzy non-decreasing fuzzy quantifier. Then, the WOW-weighting vector* $\omega = (\omega_1, \cdots, \omega_N)$ *for* $(a, \mathbf{p})$ *and* Q *is defined as follows:*

$$\omega_i = Q(\frac{\sum_{j \leq i} p_{\sigma(i)}}{\sum_{j \in L} p_{\sigma(i)}}) - Q(\frac{\sum_{j<i} p_{\sigma(j)}}{\sum_{j \in L} p_{\sigma(i)}})$$

where σ *is a permutation as above such that* $a_{\sigma(i-1)} \geq a_{\sigma(i)}$*.*

This definition permits to include the weighting of the sources to aggregation operators for categorical data. The following definition exploits this fact to define a $WOW - \mathbb{C}$ operator from an operator $\mathbb{C}$.

Definition 14 *Let* $\mathbf{X} = \{x_1, ..., x_N\}$ *be a set of information sources, let* a_i *be the value supplied by the source* x_i*, let* $\mathbb{C}$ *be an aggregation operator with* *parameter* $\mathbf{p} : X \to D$ *and let* Q *be a non-decreasing fuzzy quantifier* Q*. Then, the* $WOW - \mathbb{C}$ *operator is defined as follows:*

$$WOW - \mathbb{C}_{\mathbf{p},Q}(a_1, \cdots, a_N) = \mathbb{C}_\omega(a_1, \cdots, a_N)$$

where ω *is the WOW-weighting vector of* $(a_i, p_i)_{i=1,N}$ *and* Q *following Definition 13.*

The second aspect to be introduced in the aggregation process is compensation. This is achieved, following [1], making *data values convex.* Recall that compensation is that values $a_k \notin \{a_1, \cdots, a_N\}$ such that $min(a_i, a_j) < a_k < max(a_i, a_j)$ for $a_i, a_j \in \{a_1, \cdots, a_N\}$ can be selected. Our approach to allow compensation is to redefine the function acc in Definition 4 so that $acc(a_k) \neq 0$. In this way, a_k can be selected by the aggregation function.

Definition 15 *Let* $\mathbf{p} : X \to D \subset \mathbf{R}$ *be a weighting vector, then a mapping* $CWM_{\mathbf{p}} : L^N \to L$ *is a* Convex Weighted Median *of dimension* N *if:*

$$CWM_{\mathbf{p}}(a_1, \cdots, a_N) = a \text{ if and only if } acc'''(a) > 0.5 \geq acc'''(b)$$

where $acc'''(a) = \sum_{b \leq a} acc''(b)$, $acc''(a) = acc'(a) / \sum_{b \in L} acc'(b)$, $acc'(a) = \min(\max_{b \leq a} acc(b), \max_{b \geq a} acc(b))$, $acc(a) = \sum_{f(x_j)=a} p(x_j)$ *and where* b *is the element next to* b *in* L*. This is,* $b = \max\{x | x \in L, x < a\}$.

Now we show the application of these two procedures (the one for weighting the values and the one for allowing compensation) to the median and to the plurality rule. This application leads to the *CWOW-plurality rule.*

Definition 16 *Let* $\mathbf{p} : X \to D \subset \mathbf{R}$ *be a weighting vector, let* Q *be a non-decreasing fuzzy quantifier, then a mapping* $CWOW - Median_{\mathbf{p}} : L^N \to L$ *is a* Convex WOW-Median *of dimension* N *if:*

$$CWM_{\mathbf{w}}(a_1, \cdots, a_N) = a \text{ if and only if } acc^{iv}(a) > 0.5 \geq acc^{iv}(b)$$

where $acc^{iv}(a) = \sum_{b \leq a} acc'''(b)$, acc''' *is the WOW-weighting vector of* (L, acc'') *and* Q, $acc''(a) = acc'(a) / \sum_{b \in L} acc'(b)$, $acc'(a) = \min(\max_{b \leq a} acc(b), \max_{b \geq a} acc(b))$, $acc(a) = \sum_{f(x_j)=a} p(x_j)$ *and where* b *is the element next to* b *in* L*. This is,* $b = \max\{x | x \in L, x < a\}$.

Definition 17 *Let* $\mathbf{w}$ *be a weighting vector, and* Q *a non-decreasing fuzzy quantifier, then a mapping* $WP_{\mathbf{w}} : L^N \to \wp(L)$ *is a* CWOW-plurality rule *when* $P_{\mathbf{w}}(a_1, \cdots, a_N)$ *is the set of all those* y *in* L *so that no* z *in* L, $acc''(z) > acc''(y)$ *where* $acc''(a)$ *is the WOW-weighting vector of* (L, acc') *and* Q, $acc'(a) = \min(\max_{b \leq a} acc(b), \max_{b \geq a} acc(b))$ *and* $acc(a) = \sum_{f(x_j)=a} w(x_j)$

sources	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
values	l_4	l_1	l_0	l_2	l_1	l_0	l_4	l_5	l_1

Table 2: Information sources and values to be aggregated

	l_0	l_1	l_2	l_3	l_4	l_5	l_6	*CWOW-Med*
acc	*2*	*3*	*1*	*0*	*2*	*1*	*0*	l_4
acc'	*2*	*3*	*2*	*2*	*2*	*1*	*0*	l_2
acc"	*2/12*	*3/12*	*2/12*	*2/12*	*2/12*	*1/12*	*0*	l_2
$acc'''(\alpha = 1/8)$	*0.7993*	*0.0970*	*0.0385*	*0.0298*	*0.0245*	*0.0108*	*0.0*	l_0
$acc'''(\alpha = 1/4)$	*0.6389*	*0.1644*	*0.0705*	*0.0566*	*0.0478*	*0.0215*	*0.0*	l_0
$acc'''(\alpha = 1/2)$	*0.4082*	*0.2372*	*0.1182*	*0.1022*	*0.0914*	*0.0425*	*0.0*	l_1
$acc'''(\alpha = 1)$	*0.1666*	*0.25*	*0.1666*	*0.1666*	*0.1666*	*0.0833*	*0.0*	l_2
$acc'''(\alpha = 2)$	*0.0277*	*0.1458*	*0.1666*	*0.2222*	*0.2777*	*0.1597*	*0.0*	l_3
$acc'''(\alpha = 4)$	*0.0007*	*0.0293*	*0.0856*	*0.2006*	*0.3896*	*0.2939*	*0.0*	l_4
$acc'''(\alpha = 8)$	*0.0000*	*0.0009*	*0.0124*	*0.0867*	*0.3984*	*0.5014*	*0.0*	l_5

Table 3: The $CWOW - median$ for $\alpha \in \{1/8, 1/4, 1/2, 1, 2, 4, 8\}$

4.1 CWOW-Median

In this section, we study the *CWOW-Median* procedure defined in Definition 16. We begin giving an example that shows the suitability of the approach for obtaining, with appropriate parameterizations, values between the minimum and the maximum of the value to be aggregated. Then we analyze the properties of the operator focusing in the monotonicity condition.

Example 1 *Let $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$ be a set of information sources, let $f(x_i) = a_i$ be defined as in Table 2 (here $L = \{l_0, l_1, l_2, l_3, l_4, l_5, l_6\}$), and let $p(x_i) = 1$ for all x_i, then, the $CWOW - Median$ for $Q(x) = x^\alpha$ for $\alpha \in \{1/8, 1/4, 1/2, 1, 2, 4, 8\}$ is given in Table 3. This table includes the computed vectors acc, acc$'$, acc$''$ that are common for all $CWOW - Median$ operators and then the vector acc$'''$ for each considered α. The last row of the column describes the aggregated values $CWOW - Median$ for each of the values and also the median for the original data (first row – acc row) and for the convex weighted median (second and third row, denoted by acc$'$ and acc$''$ rows).*

This example shows that the $CWOW - Median$ permits to overcome the compensation inconvenience faced by the original median operator. Note that it is possible to obtain l_3 as the output when $\alpha = 2$ while l_3 was not one of the values to be aggregated. It can also be observed that the operator, by means of the α parameter, permits to obtain values between the minimum and the maximum of the a_i. In our case, the function moves from l_0 to l_5. Moreover, the function cannot result into values larger than the maximum of the a_i or smaller than the minimum of the a_i. This fact also implies that the operator satisfies unanimity (if all sources agree in a value l_i, the outcome is this very value l_i)

Proposition 1 *$CWOW-Median$ is an aggregation operator satisfying:*

1. *$min(a_1, \cdots, a_N) \leq CWOW - Median(a_1, \cdots, a_N) \leq \max(a_1, \cdots, a_N)$*
2. *Unanimity $CWOW - Median(l, l, \cdots, l) = l$ for all $l \in L$*

Nevertheless, this operator presents a drawback. The following proposition establishes this negative property.

Proposition 2 *The $CWOW-Median$ does not satisfy monotonicity. This is, it does not hold*

$$CWOW - Median(a_1, \cdots, a_N) \leq CWOW - Median(a'_1, \cdots, a'_N)$$

for some $a_i \leq a'_i$ where $i \in \{1, \cdots, N\}$

Non-monotonicity is a consequence of the fact of making the function *acc* convex. Augmenting the values of *acc* for all the elements below the previous median value can violate monotonicity. This is illustrated in the following example:

Example 2 *Let us consider 16 information sources $X = \{x_1, x_2, \cdots, x_{16}\}$ giving information over a set L of 11 ordered categories $L = \{l_0, l_1, \cdots, l_{10}\}$. The information supplied by the sources is as follows: 6 of the sources supply the value l_0 and the other 10 supply the values $l_1, l_2, \cdots, l_{10}$. This is,*

$$a = (l_0, l_0, l_0, l_0, l_0, l_0, l_1, l_2, \cdots, l_9, l_{10})$$

To aggregate this values, the $CWOW - Median$ is used. The corresponding acc function is given in the first row of Table 4. The application of the simple median to these values is given in the last column of the first row. The second and the third row of this table gives the acc' and acc'' functions. This is, the convex function and the normalized function (the one that add to one). The last column of these rows shows the value of the $CWOW - Median$: l_3.

Let us now consider that one of the sources that supplied the category l_0 (say x_1) changes the value by l_2. The corresponding a' vector is now:

$$a' = (l_2, l_0, l_0, l_0, l_0, l_0, l_1, l_2, \cdots, l_9, l_{10})$$

Note that this vector is monotonic increasing in relation to the previous vector a because, $a'_i \geq a_i$ for all $i \in \{1, \cdots, N\}$.

The corresponding acc function is given in the fourth row of Table 4. The last column of this row gives the Median of the values. The median is a monotonic function and it can be seen that in this case the final value is not modified by the change of l_0 by l_2. In the last two columns of this table, functions acc' and acc'' are displayed. The last column in the rows give the result for the $CWOW - Median$ function: l_2.

The example shows that monotonicity is not sastisfied because changing the value $a_1 = l_0$ by $a'_1 = l_2$ (and keeping all the others $a'_i = a_i$), the outcome of the function is l_2 instead of l_3 and thus violates the equation:

$$CWOW - Median(a_1, \cdots, a_N) \leq CWOW - Median(a'_1, \cdots, a'_N)$$

The violation of the monotonicity condition is due to several factors (see Table 4): (i) the replacement of the value l_0 by two values instead of one in acc', and thus incrementing the number of total values in the median from 16 to 17 (see denominators in rows acc''); (ii) the two additional values are lesser than l_3 and thus decrements the final outcome (note different values in columns l_1 and l_2 in rows acc'). Both factors are caused by the process of making acc' a convex function (in fact, incrementing the number of values l_i smaller than l_3). Note that for the original Median function, the final aggregated value is not modified (the function is indeed monotonic).

	l_0	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}	CWOW-Med
acc	*6*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	l_3
acc'	*6*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	l_3
acc"	*6/16*	*1/16*	*1/16*	*1/16*	*1/16*	*1/16*	*1/16*	*1/16*	*1/16*	*1/16*	*1/16*	l_3
acc	*5*	*1*	*2*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	l_3
acc'	*5*	*2*	*2*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	l_2
acc"	*5/17*	*2/17*	*2/17*	*1/17*	*1/17*	*1/17*	*1/17*	*1/17*	*1/17*	*1/17*	*1/17*	l_2

Table 4: Example of non-monotonicity for the $CWOW - Median$

.

Nevertheless, although these examples do not satisfy the monotonicity condition, it is clear that variations on the result are small (one label is changed by a contiguous one) and can be accepted from the point of view that we are using ordinal scales with no established semantics. In fact, the violation of the monotonicity condition is found when for a category l_i the acc''' function ($acc'''(l_i)$) is near the cutting point 0.5. Note that $acc'''(l_2) = 0.5$ and $acc'''(l_3) = 0.5625$ for the a vector, and that $acc'''(l_2) = 0.5294118$ for a'. On the light of ordinal scales as scales with some uncertainty (e.g. imprecision or fuzzy terms), we can understand non-monotonicity results as errors in the limits of the meaning of the category.

It has to be said that in general, it is possible to find examples of non-monotonicity in which replacing a value l_a by a larger value l_b results in a large change of the outcome. However, this requires a set L with a large number of categories and a large set of sources, a situation that is not common when dealing with ordinal scales (specially, the case of having a large set of categories). This is illustrated in the following example:

Example 3 *Let us consider a set X consisting on 1006 information sources, each supplying a value in the ordinal scale $L = \{l_0, l_1, l_2, \cdots, l_{1000}\}$. Let x_i*

supply the value l_i for $i = 1, \cdots, 1000$ and let $x_{1001}, \cdots, x_{1006}$ supply the value l_0. Then, the $CWOW - Median$ of these values is l_{498}.

Let us consider now that x_{1001} replaces the value l_0 by the value l_{250}, then, the $CWOW - Median$ is l_{373}.

As $l_{250} > l_0$, but $l_{373} < l_{498}$, the monotonicity condition is not satisfied. In this case, the number of categories between the original value and the new one is large but the number of categories in the set L is also very large.

To have a better understanding of the situations in which the $CWOW - Median$ violates monotonicity (this understanding is required to apply the aggregation operator properly), we have studied in detail different situations and analyzed them to know whether the operator satisfies monotonicity or not.

We have considered two different scenarios and randomly generated several instantiations. In each instantiation, two monotonic vectors $a^1 = (a^1_1, \cdots, a^1_N)$ and $a^2 = (a^2_1, \cdots, a^2_N)$ (i.e., $a^1_i \leq a^2_i$) were generated and the $CWOW - Median$ was applied to them. Monotonicity was then checked.

In both scenarios, we consider an ordinal scale consisting on l categories ($l = R + 1$ using the notation $L = \{l_0, l_1, \cdots, l_R\}$ used so far), N information sources and that the difference between vectors a^1 and a^2 is that K-information sources have changed their value in a^1 by a larger one in a^2 (this is, $|\{a_i | a^1_i \neq a^2_i\}| \approx K$). For each scenario, m random instantiations have been considered.

The two scenarios studied are the following ones:

1. All the sources changing a value in a^1 to another one in a^2 had the same value in a^1 and change to the same value in a^2. This is, for all i such that $a^1_i \neq a^2_i$, $a^1_i = \alpha$ and $a^2_i = \beta$. In this case, if for a given parameterization K, the number of categories $a^1_i = \alpha$ is K' wit K' less than K, only K' sources will change their value.

2. Sources that change their values can have different values both in a^1 and in a^2.

According to all this, for each of the scenarios, an example is defined according to four parameters (l, N, K, m). For evaluating the aggregation function, we have considered the following parameters for the two scenarios:

- The number of categories: $l = 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 50, 100$
- The number of sources: $N = 5, 10, 15, 20, 25, 30, 40, 50, 75, 100, 200, 500, 1000$
- The number of changed values: $K = 2, 3, 4, 5, 10, 100$

Experiments were run either 1000 or 10000 times ($m = 1000$ or $m = 10000$). The results of the experiments are displayed in Tables 5 – 17. Tables show the number of cases that violate monotonicity. This is, each cell of the table indicates how many times the monotonicity condition was violated when m experiments were executed.

These experiments were programmed in CLisp (running on RedHat 6.2 for a PC) and for scenario 1 with $K = 2$ and $m = 10000$ it took 3 hours to compute all

$l \backslash N$	5	10	15	20	25	30	40	50	75	100	200	500	1000
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	23	17	5	3	2	1	1	0	0	0	0	0	0
5	19	43	28	25	29	21	12	13	3	0	1	0	0
6	34	69	53	34	37	22	26	18	15	13	15	7	5
7	37	104	93	80	66	52	48	34	34	16	4	0	0
8	39	108	105	103	79	74	45	38	26	17	15	10	1
9	41	168	136	131	105	100	75	66	52	46	14	3	0
10	27	146	133	132	98	141	77	70	49	38	28	22	16
20	20	211	214	185	192	216	183	183	127	131	69	28	20
30	4	143	212	191	196	235	236	234	188	179	104	59	28
50	2644	3592	3617	3789	3750	3714	3766	3728	3667	3677	3632	1933	1115
100	0	27	220	697	1550	2695	5230	6649	6934	6846	6886	3168	111

Table 5: Experiments for scenario 1 with $K = 1$ (number of changed values from a^1 to a^2). Rows correspond to different number of labels (parameter l) and columns correspond to different number of information sources (parameter N). 10000 tests have been performed for each experiment
.

examples for all considered pairs of N and l (this is the completion of Table 6). Instead, the computation of the Table 17 (scenario 2, $K = 100$ and $m = 10000$) took about 6 hours.

From the tables, it can be observed that for a small number of categories the number of monotonicity violations is small (less than 3%). This number is even smaller for the second scenario. The experiments also show that for the second scenario when the value K increases, the percentage of violations decreases (specially for the experiments with a small number of categories – see for example Tables 16 and 17 and compare with Table 12). For the first scenario, conclusions are not so clear, but it seems that larger values of K, the number of violations decreases for a small number of information sources and increases for a larger number of sources. For example, for $K = 1$ and $N = 10$ and $N = 15$, the cells for $l = 10$ are about 140 while for $K = 5$, the same cells are about 100, for $K = 20$ they are about 85, for $K = 100$ they are also about 85. Instead, the corresponding cells ($l = 10$) for $N = 40$ and $N = 50$ are, respectively, for $K = 1$ about 75, for $K = 5$ about 148, for $K = 10$, 127 and 164, for $K = 100$, 140 and 178. Thus, the number of violations tends to decrease for a small number of sources.

Worst cases are found for large number of categories (30 or larger). In this case, a small variation of the input data implies a large modification of the convex function (this is the case of Example 3). For example, Table 5 shows that for 50 categories and 5 information sources there is 26.44% percent of the cases that do not satisfy monotonicity, for 100 categories and 75 sources we have

$l\backslash N$	5	10	15	20	25	30	40	50	75	100	200	500	1000
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	15	26	5	2	2	3	0	0	0	0	0	0	0
5	17	38	29	28	30	27	26	19	7	3	0	0	0
6	31	54	58	67	55	37	33	28	16	11	28	7	15
7	23	72	91	70	79	81	58	62	35	27	6	0	0
8	41	62	84	101	92	106	68	76	54	43	37	20	7
9	21	90	110	106	133	109	112	94	84	65	31	2	0
10	27	94	111	118	146	137	99	104	82	73	39	19	16
20	14	112	109	146	206	218	213	224	185	182	111	68	35
30	6	74	97	113	181	215	246	269	223	257	177	92	51
50	734	1010	2922	3625	3729	3705	3730	3770	3744	3710	3809	3688	2318
100	0	0	3	3	9	12	50	123	1949	6228	6905	6647	616

Table 6: Experiments for scenario 1 with $K = 2$. Rows correspond to number of labels and columns correspond to number of information sources. 10000 tests have been performed for each experiment

.

almost 70% percent of the cases.

From the point of view of aggregation for prototype selection, the experiments show that the proposed aggregation method is a valid alternative because the usual number of categories is usually smaller than 15. For example, in the experiments in [3], the average number of categories is 13, only 30% of the variables have more than 15 categories and the variable with a larger number has 25 categories. In addition, in the particular case of clustering for microaggregation [3], [4], the number of values to be aggregated is usually below 10. In a general clustering problem, this number will be quite larger but the number of categories will be about the same.

An additional element to be taken into account is that in our experiments the values are generated randomly and, thus, a given vector a can have very dissimilar values. However, when applying aggregation to clustering, the values would be similar. In fact, they have to be so because they are put together in the same cluster because they are similar. The effects of non monotonicity would be smaller in this latter case. Recall that non monotonicity is caused by the introduction of new elements in the convex function acc', therefore, when values are similar, the number of added elements will be small.

5 Conclusions

In this work we have reviewed existing aggregation operators in ordinal scales for their application to prototype construction. We have analysed their drawbacks and we have proposed two general procedures to solve them. Then, we have applied these procedures to the median to define the $CWOW - Median$. We have analyzed some of the properties of this operator. We have seen that it satisfies unanimity and that the value belongs to the interval defined by the minimum and maximum of the values. We have shown that the procedure does not satisfy monotonicity. We have shown with an example that the modification of a single label does not modify in a substantial way the outcome (a label is changed by the contiguous one). Only for an example with a large domain L, the outcome of the $CWOW - Median$ is modified substantially. Experiments have confirmed that violations of monotonicity are not relevant for a small number of categories and of sources. This is the typical case in clustering and more specially in microaggregation. Experiments show that monotonicity is not satisfied for a large proportion of scenarios when the number of categories is large. However, this is not a common situation.

As in usual applications the number of categories in L is not large, and the non-monotonicity can be understood from the point of view of the uncertainty attached to categories (e.g. imprecision), we consider appropriate the use of *CWOW-Median* for prototype selection. In particular, because it allows compensation a property that the other operators lack, and also because it allows a parametric definition (through the quantifier) that allows the user to customize the application or to apply learning procedures. In particular, and as shown in [3], parameterization is a relevant aspect in microaggregation to find the best tradeoff between information loss and the risk of releasing unprotected data.

$l \backslash N$	5	10	15	20	25	30	40	50	75	100	200	500	1000
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	23	25	6	4	3	1	1	0	0	0	0	0	0
5	23	54	36	45	41	37	34	24	6	1	0	0	0
6	20	47	83	50	61	56	34	36	32	25	25	15	11
7	23	67	84	90	100	97	89	71	60	43	14	1	0
8	25	75	73	89	106	95	81	84	55	54	49	32	20
9	20	80	106	105	129	116	127	113	98	68	36	3	1
10	28	83	105	130	104	136	124	135	96	85	81	39	26
20	11	98	77	121	177	159	181	205	219	201	130	85	52
30	10	55	103	99	131	185	208	228	249	272	225	110	63
50	722	322	1091	2263	3156	3524	3706	3690	3755	3672	3725	3787	3325
100	0	0 0	0	4	3	17	15	17	32	209	6882	6866	2766

Table 7: Experiments for scenario 1 with $K = 3$. Rows correspond to number of labels and columns correspond to number of information sources. 1000 tests have been performed for each experiment

$l\backslash N$	5	10	15	20	25	30	40	50	75	100	200	500	1000
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	23	25	7	9	5	2	0	0	0	0	0	0	0
5	25	40	30	51	30	49	31	25	6	7	0	0	0
6	25	59	58	72	55	60	48	54	46	35	40	20	13
7	20	70	72	66	85	105	80	86	63	45	9	0	0
8	25	69	80	84	102	109	104	110	75	79	37	31	25
9	29	82	110	110	133	130	135	137	102	82	44	3	0
10	26	88	95	114	112	131	163	111	131	95	53	52	43
20	11	93	101	123	177	186	196	212	219	217	154	87	62
30	7	65	110	103	141	162	168	247	292	289	261	142	98
50	740	278	158	322	950	1936	3517	3711	3728	3864	3843	3754	3727
100	1	0	0	3	2	11	17	11	29	26	3988	6839	5879

Table 8: Experiments for scenario 1 with $K = 4$. Rows correspond to number of labels and columns correspond to number of information sources. 10000 tests have been performed for each experiment

Acknowledgments

Partial support of the European Community under the contract CASC (IST-2000-25069) and of the CICYT under the contract STREAMOBILE (TIC2001-0633-C03-01/02) is acknowledged.

References

[1] Bardossy, A., Duckstein, L., Bogardi, I., (1993), Combination of fuzzy numbers representing expert opinions, Fuzzy sets and systems, 57 173-181.

[2] Choquet, G., (1968), Theory of Capacities, Ann. Inst. Fourier 5 (1968) 131-296.

[3] Domingo-Ferrer, J., Torra, V., (2001), A Quantitative Comparison of Disclosure Control Methods for Microdata, 111-133 in Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies, P. Doyle, J. I. Lane, J. J. M. Theeuwes, L. M. Zayatz (Eds.), Elsevier.

[4] Domingo, J., Torra, V., (2002), Aggregation techniques for statistical confidentiality, in "Aggregation operators: New trends and applications", (Ed.), R. Mesiar, T. Calvo, G. Mayor, Physica-Verlag, Springer, in press.

[5] Dubois, D., Prade, H., (1986), Weighted minimum and maximum operations, Information Sciences, 39 (1986) 205-210.

$l\backslash N$	5	10	15	20	25	30	40	50	75	100	200	500	1000
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	14	31	4	11	0	1	1	0	0	0	0	0	0
5	13	44	40	42	40	39	41	21	21	6	0	0	0
6	25	52	72	65	70	76	72	70	49	42	36	28	23
7	31	65	74	93	84	83	97	101	78	50	17	0	0
8	30	90	79	114	125	101	115	105	86	75	56	48	28
9	23	79	103	99	125	126	155	148	97	96	44	9	0
10	27	87	104	115	146	124	148	148	115	116	80	45	35
20	14	106	116	135	137	155	179	221	237	253	175	98	81
30	5	64	121	114	140	183	188	212	234	298	273	179	100
50	697	243	123	103	323	735	2033	3114	3692	3773	3835	3731	3709
100	0	0	1	6	7	13	13	13	15	31	202	6951	6866

Table 9: Experiments for scenario 1 with $K = 5$. Rows correspond to number of labels and columns correspond to number of information sources. 10000 tests have been performed for each experiment

[6] Fishburn, P.C., Rubistein, A., (1986), Aggregation of equivalence relations, J. of classification, 3 61-65.

[7] Godo, L., Torra, V., (2000), On aggregation operators for ordinal qualitative information, IEEE Trans. on Fuzzy Systems, 8:2 (2000) 143-154.

[8] Godo, L., Torra, V., (2001), Extending Choquet Integrals for Aggregation of Ordinal Values, Int. J. of Uncertain Fuzziness and Knowledge-Bases, in press.

[9] Grabisch, M., Nguyen, H. T., Walker, E. A., (1995), Fundamentals of Uncertainty Calculi with Applications to Fuzzy Inference, Kluwer Academic Publishers, Dordreht, The Netherlands.

[10] Herrera, F., Herrera-Viedma, E., (1997), Aggregation Operators for Linguistic Weighted Information. IEEE Transactions on Systems, Man and Cybernetics, 27 646-656

[11] Herrera, F., Herrera-Viedma, E., Verdegay, J.L., (1995), A Sequential Selection process in Group Decision Making with a Linguistic Assessment Approach, Information Science, 85 (1995) 223-239.

[12] Keller, A., Klawonn, F., (2000), Fuzzy Clustering with Weighting of Data Variables, Int. J. of Uncertain Fuzziness and Knowledge-Based Systems 8:6.

[13] Marichal, J.-L., (1999), Aggregation operators for multicriteria decision aid, PhD Dissertation, Faculte des Sciences, Universite de Liege, Liege, Belgium.

$l\backslash N$	5	10	15	20	25	30	40	50	75	100	200	500	1000
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	14	27	14	5	2	1	1	0	0	0	0	0	0
5	25	42	24	54	46	52	47	42	17	11	1	0	0
6	30	63	71	78	85	73	106	102	93	83	57	38	31
7	29	65	87	89	88	116	107	114	95	73	18	1	0
8	28	72	84	88	119	97	119	147	156	151	82	71	54
9	24	80	82	122	112	131	148	141	158	182	67	11	1
10	31	83	85	97	111	155	127	164	172	175	126	79	66
20	13	82	99	136	150	158	183	191	248	259	295	180	90
30	6	74	96	116	133	160	195	225	261	315	414	315	171
50	744	256	117	68	52	40	31	49	310	1744	3790	3941	3779
100	0	0	0	5	5	13	18	18	25	28	47	458	6902

Table 10: Experiments for scenario 1 with $K = 10$. Rows correspond to number of labels and columns correspond to number of information sources. 10000 tests have been performed for each experiment

[14] Mayor, G. Torrens, J., (1993), On a Class of Operators for Expert Systems, Int. J. of Intelligent Systems, 8, 771-778.

[15] Roberts, F. S., (1991), On the indicator function of the plurality function, Mathematical Social Sciences, 22 163-174.

[16] Sugeno, M., (1974), Theory of Fuzzy Integrals and its Applications. (PhD Dissertation). Tokyo Institute of Technology, Tokyo, Japan.

[17] Torra, V., (1996), Negation functions based semantics for ordered linguistic labels, Int. J. of Intelligent Systems, 11 975-988.

[18] Torra, V., (1997), The Weighted OWA operator, International Journal of Intelligent Systems, 12 153-166.

[19] Torra, V., (2001), Aggregation of linguistic labels when semantics is based on antonyms, Int. J. of Intel. Syst., 16:4 (2001) 513-524.

[20] Yager, R. R., (1988), On ordered weighted averaging aggregation operators in multi-criteria decision making, IEEE Trans. on SMC, 18 (1988) 183-190.

[21] Yager, R. R., (1996), Quantifier Guided Aggregation Using OWA operators, Int. J. of Int. Systems, 11 (1996) 49-73.

[22] Yager, R. R., (2000), On the Control of Hierarchical Clustering, Proc. IPMU 2000, Madrid, Spain, 1335-1341.

$l\backslash N$	5	10	15	20	25	30	40	50	75	100	200	500	1000
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	26	24	11	4	3	1	0	0	0	0	0	0	0
5	18	40	19	46	43	56	45	42	18	13	0	0	0
6	31	67	80	76	78	81	84	93	129	152	222	230	265
7	26	58	95	93	97	107	95	121	123	108	102	71	10
8	22	65	68	90	102	106	125	150	172	165	257	375	341
9	30	70	95	105	150	134	144	164	160	191	164	173	103
10	25	84	99	123	118	147	140	178	181	199	256	385	474
20	13	113	118	123	141	150	179	196	271	304	308	374	466
30	6	67	100	105	146	178	196	239	251	330	407	453	437
50	684	257	98	68	39	26	40	52	73	71	118	202	322
100	0	3	0	5	6	7	11	19	26	28	42	33	16

Table 11: Experiments for scenario 1 with $K = 100$. Rows correspond to number of labels and columns correspond to number of information sources. 10000 tests have been performed for each experiment

$l\backslash N$	5	10	15	20	25	30	40	50	75	100	200	500	1000
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	30	6	3	0	0	1	0	0	0	0	0	0	0
5	20	13	8	9	7	13	8	8	2	4	0	0	0
6	26	33	29	18	21	8	10	10	6	7	4	5	2
7	28	76	59	36	33	33	32	22	16	21	4	0	0
8	36	103	73	68	47	40	30	15	19	16	12	11	8
9	48	153	100	87	94	70	62	43	41	35	19	0	0
10	58	158	122	114	93	80	74	59	61	31	20	26	11
20	27	303	314	282	318	262	252	239	170	147	92	42	31
30	11	209	346	331	328	364	329	341	314	250	161	75	47
50	3392	4476	5033	5024	4949	4697	4514	4271	3864	3562	3152	2308	1410
100	1	15	131	367	783	1277	2318	2883	3591	6257	5910	4874	458

Table 12: Experiments for scenario 2 with $K = 2$. Rows correspond to number of labels and columns correspond to number of information sources. 10000 tests have been performed for each experiment

$l\backslash N$	5	10	15	20	25	30	40	50	75	100	200	500	1000
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	22	30	15	8	0	2	0	1	0	0	0	0	0
5	26	43	42	36	27	33	32	17	9	6	0	0	0
6	25	68	61	69	49	59	41	34	41	18	22	21	9
7	29	63	67	105	94	80	82	71	57	44	15	0	0
8	27	76	84	108	105	93	87	73	59	61	39	34	26
9	26	90	94	122	116	123	113	118	96	70	43	2	0
10	29	85	95	112	137	132	134	123	93	54	54	39	38
20	18	77	97	145	156	168	204	249	224	214	128	74	50
30	4	68	119	115	146	177	191	227	268	305	237	135	71
50	688	337	1069	2251	3178	3548	3733	3776	3790	3713	3760	3826	3454
100	0	0	3	3	2	6	11	18	33	202	6891	6910	2707

Table 13: Experiments for scenario 2 with $K = 3$. Rows correspond to number of labels and columns correspond to number of information sources. 10000 tests have been performed for each experiment

$l\backslash N$	5	10	15	20	25	30	40	50	75	100	200	500	1000
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	11	0	0	0	0	0	0	0	0	0	0	0	0
5	10	4	1	1	1	1	0	0	0	0	1	0	0
6	13	10	8	3	5	3	5	0	3	1	3	1	3
7	19	28	16	9	10	5	10	12	5	3	2	0	0
8	30	47	27	17	14	12	6	9	5	6	10	4	3
9	33	65	60	30	37	28	28	25	23	17	7	3	0
10	52	110	56	52	46	49	32	28	20	16	18	6	9
20	54	405	341	304	252	249	234	208	167	142	80	42	25
30	22	317	463	423	460	446	384	368	323	292	177	95	57
50	3224	4136	5440	5899	5981	5856	5569	5104	4530	4276	3678	2833	1930
100	1	6	48	110	254	482	1078	1499	2194	3961	6682	6614	2566

Table 14: Experiments for scenario 2 with $K = 4$. Rows correspond to number of labels and columns correspond to number of information sources. 10000 tests have been performed for each experiment

$l\backslash N$	5	10	15	20	25	30	40	50	75	100	200	500	1000
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	9	0	1	0	0	0	0	0	0	0	0	0	0
5	7	2	1	4	1	0	0	0	1	1	1	0	0
6	7	2	1	3	0	3	0	0	0	1	0	1	2
7	23	30	5	2	7	8	1	6	6	5	2	1	0
8	26	28	19	7	11	7	1	9	6	4	3	4	1
9	30	47	30	27	20	17	22	13	8	10	6	1	0
10	41	59	38	33	29	14	17	16	13	14	5	4	6
20	52	357	295	317	261	208	199	186	136	129	79	45	26
30	43	356	496	443	462	439	415	383	305	256	175	90	60
50	3012	3727	5261	5912	6029	6044	5847	5496	4798	4349	3798	3036	2156
100	0	4	28	60	164	324	704	1091	1686	3032	6647	7012	3374

Table 15: Experiments for scenario 2 with $K = 5$. Rows correspond to number of labels and columns correspond to number of information sources. 10000 tests have been performed for each experiment

$l\backslash N$	5	10	15	20	25	30	40	50	75	100	200	500	1000
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	2	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0	0	0	0
6	1	1	0	0	0	0	0	0	0	0	0	0	0
7	3	2	0	0	1	0	0	0	0	0	0	0	0
8	1	3	0	1	1	0	1	2	0	1	0	0	0
9	10	8	5	1	1	1	2	0	1	1	2	0	0
10	21	14	10	5	2	2	3	3	3	1	2	2	2
20	53	284	148	169	128	120	62	69	48	50	42	23	21
30	41	405	437	353	337	332	269	242	208	163	123	66	41
50	2131	2121	3429	4569	5242	5809	6140	6009	5380	4970	4116	3359	2624
100	4	0	0	3	7	24	76	151	393	970	3301	8344	6666

Table 16: Experiments for scenario 2 with $K = 10$. Rows correspond to number of labels and columns correspond to number of information sources. 10000 tests have been performed for each experiment

$l \backslash N$	5	10	15	20	25	30	40	50	75	100	200	500	1000
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	4	1	0	1	0	0	0	0	0	0	0	1
50	416	286	91	39	20	10	13	83	954	2860	4747	3406	3093
100	5	0	0	0	0	0	0	0	0	0	0	0	21

Table 17: Experiments for scenario 2 with $K = 100$. Rows correspond to number of labels and columns correspond to number of information sources. 10000 tests have been performed for each experiment

Extending Microaggregation Procedures using Defuzzification Methods for Categorical Variables

Josep Domingo-Ferrer, Vicenç Torra

Abstract— **Defuzzification is one of the fundamental steps in the development of fuzzy knowledge based systems. Given a fuzzy set μ over the reference set X, defuzzification applied to μ returns an element of X. While a large number of methods exists for the case of X being a numerical scale, only few methods are applicable when X corresponds to a categorical scale.**

Aggregation procedures have been extensively used in defuzzification in numerical scales. This is so because defuzzification has been studied as equivalent to the computation of an expected value. In this work we present the reversal approach, we study defuzzification procedures for their application to aggregation. We focus on the development of defuzzification methods for the case of X being an ordinal scale. This is, X is a set of finite values in which a total order is defined. Our ultimate goal is to apply these methods to microaggregation (a Statistical Disclosure Risk).

Index Terms— **Defuzzification, ordinal scales, aggregation procedures, selection procedures.**

I. Introduction

Fuzzy knowledge based systems (see [1] and [2] for details) are one of the most successful applications of fuzzy sets [3]. These systems are rule based systems in which the predicates of the antecedents and consequents are defined in terms of fuzzy sets. This is, given a fuzzy rule of the form

$$\textbf{if}\ \ X\ is\ A\ \textbf{and}\ \ \ldots\textbf{and}\ \ \ Y\ is\ B\ \textbf{then}\ \ Z\ is\ C$$

X, Y and Z are variables and A, B and C are fuzzy sets on the reference sets of X, Y and Z. We will denote the reference set of a variable X by D_X. For example, if X corresponds to temperature, A can correspond to a fuzzy set describing values near zero degrees, in this case, the reference set D_X corresponds to real numbers or a subset of them ($D_X = \mathbb{R}$).

Given a piece of information on the variables of the antecedent (their value or a fuzzy set describing possible values), the information is propagated into the conclusion to get information about the possible values of the variable in the conclusion. In the case above, given values for variables $X, \cdots, Y$, a system would obtain possible values for Z. These values are usually described by means of a fuzzy set on the reference set of Z (i.e., D_Z). If no information can be inferred, the set of possible values for Z is usually set equal to the empty set. In fact, this corresponds to disjunctive systems (conjunctive systems make, in this case, the conclusion equal to the whole set [1]). In this way, the conclusion of each rule takes into account available information (the values of the variables) and the relevance of this information to the rule at hand (in which degree the antecedent is fulfilled).

Most fuzzy knowledge based systems correspond to one stage fuzzy systems. These systems are defined by means of a flat set of rules, all having the same set of variables in the antecedent and the same variables in the consequence. Then, in a given instant of time, all rules are applied and the conclusions of all rules are combined to build the conclusion of the system. This combined conclusion corresponds to a fuzzy set (we will denote this combined fuzzy set by $\mu_{\mathbb{C}}$). A final step of the system is defuzzification. This is to transform the fuzzy set $\mu_{\mathbb{C}}$ into a value in the reference set of the conclusion variable. For the example above, this is: $defuzzification(\mu_{\mathbb{C}}) \in D_Z$

When studying defuzzification methods an important aspect to be taken into account is the scale of the reference set. Three main types can be distinguished:

Numerical: The conclusion is either a real number or a natural number.

Ordinal scale: The conclusion is categorical (e.g., linguistic labels) and values are ordered (there is a total relation over the categories).

Nominal scale: The conclusion is categorical but there is no ordering relation between categories.

While most control systems fall into the first type, most configuration systems fall in the second and third type (e.g., the one in [4]).

At present there exists a large number of defuzzification procedures for numerical scales, while existing methods for ordinal scales are more limited. In this work we study defuzzification methods for ordinal scales. Our interest in these methods is for their application to the aggregation of values (fusion of information) in these scales.

In fact, usually the relationship between aggregation (fusion) and defuzzification is observed in the other direction. This is so because, two main approaches are considered in defuzzification (this is explained in more detail in the rest of the paper):

Element selection: From the reference set of the conclusion variable, one element has to be selected. This selection has to be based on the combined fuzzy set. In the example above, this means to select a value from the set $\mu_{\mathbb{C}}$ for the variable Z in the reference set D_Z.

Aggregation: The defuzzified value corresponds to the aggregation of the available information. This can be seen as

J. Domingo-Ferrer is with the Department of Computer Engineering and Maths - ETSE, Universitat Rovira i Virgili, Avga Països Catalans 26, E-43007 Tarragona, Catalonia. E-mail: jdomingo@etse.urv.es

V. Torra is with the Institut d'Investigació en Intel·ligència Artificial - CSIC, Campus UAB s/n, E-08193 Bellaterra, Catalonia. E-mail: vtorra@iiia.csic.es

equivalent to returning an expected value of the combined fuzzy set.

According to the second approach, almost all aggregation procedures can be used for defuzzification. This is particularly interesting in the case that the variables are defined in numerical scales due to the large number of aggregation procedures. Also, the selection of an appropriate procedure makes possible the customization of an application. In the case of variables on ordinal or numerical scales, this approach is not as appropriate because there are only a few applicable aggregation procedures. Nevertheless, the relationship can be studied in the reversal direction to develop new aggregation procedures, and at the same time extending defuzzification methods.

II. Aggregation Procedures in Ordinal Scales

The development of aggregation procedures for ordinal scales is briefly reviewed in [5]. A more detailed review that also analyzes the properties of the procedures in the view of prototype selection in clustering is given in [6]. It has to be noted that, in some sense, the properties of a defuzzification method are similar to the ones of the methods for selecting the prototype of a class and, thus, the analysis in that work is relevant here.

In the rest of the paper we assume that the values to be aggregated (or the domain X in the case of defuzzification) are the following ones: $L = \{l_0, \cdots, l_R\}$ where $l_0 \leq_L l_1 \leq_L \cdots \leq_L l_R$.

Aggregation procedures in ordinal scales can be classified into three main classes according to the semantics underlying the categories (or linguistic labels) in L:

Explicit quantitative or fuzzy scales: There is a function that assigns to each value l_i a value in a numerical (or fuzzy scale). Aggregation operators can be defined on L according to this translation function. An operator of this class is described in [5].

Implicit numerical scale: In this class of operators the translation is not explicitly defined but it is implicitly assumed. Two operators of this class are the Linguistic OWA [7] and the Linguistic WOWA [8].

Operating directly on qualitative scales: The operators of this class do not assume the existence of any translation function either implicitly or explicitly. Operators are only based on operators that can be directly defined in the ordinal scale. Example of operators that can be defined in the ordinal scale are the *minimum* and the *maximum* (that rely on the relation $\leq_L$) and also t-norms and t-conorms (because these operators can be defined axiomatically – see [9]).

When we restrict to the third class, the number of existing operators is small. In fact, four main families can be distinguished:

Plurality rule: This function selects the set of most frequent elements. Weighted plurality rule can also be defined. In this case, the *most frequent* elements are the ones that accumulate *larger weights.*

Median and order statistics: This function selects the element that occupies the central position when all the elements to be aggregated are ordered. A weighted median can also be defined. We classify *order statistics* procedures in this class, because they correspond to the selection of those elements that occupy other positions (e.g., first, second or last) than the central one.

Sugeno integral: This integral can be interpreted as an aggregation operator. In this case, values are aggregated taking into account a fuzzy measure. This fuzzy measure is used to measure the importance of the information sources (the sources that supply the values to be aggregated). It is well known that the Sugeno integral generalizes several other aggregation operators as the weighted minimum and the weighted maximum.

Ordinal weighted mean: This operator is the ordinal counterpart of the weighted mean. The definition of this operator tries to mimic the definition of the numerical weighted mean. Instead of addition and product, it uses t-conorms and t-norms. Ordinal Choquet integral has also been defined using the same approach.

Some of the difficulties underlined in [6] for the application of these operators to prototype selection are the following ones:

1. *Difficulty for defining the parameters by non experienced users:* Operators like the Sugeno integral and the Ordinal weighted mean that need that someone defines fuzzy measures or t-norms/t-conorms are not appropriate because these parameters are not easy to define.
2. *Inconvenience of non-compensative operators:* For most reviewed operators, the result of the aggregation should be one of the values to be aggregated. This means that the average of a small and a large value cannot be a value somewhere in between. Instead, compensation is allowed in the numerical case. E.g., the aggregation of 0 and 1 is 0.5 when the aggregation operator is the arithmetic mean.
3. *Difficulty of defining parametric operators:* Parameterization of existing operators is difficult. Some of the operators require weights in ordinal scales. Fuzzy measures are even more difficult to define because they are defined over parts of the sources and, here, the only information is the membership value for each element.

Taking all these aspects into account it seems that the most appropriate aggregation procedure for prototype selection is the median. However, it does not allow for compensation and does not include any parameterization. To overcome these two difficulties, the CWOW-Median was defined in [6]. Its definition is as follows:

Definition 1: Let $\mathbf{p} : \mathbf{X} \rightarrow \mathbf{D} \subset \mathbf{R}$ be a weighting vector, let Q be a non-decreasing fuzzy quantifier, then a mapping $CWOW - Median_{\mathbf{p}} : L^N \rightarrow L$ is a *Convex WOW-Median* of dimension N if:

$$CWM_{\mathbf{w}}(a_1, \cdots, a_N) = a \text{ iff } acc'''(a) > 0.5 \geq acc'''(b)$$

where acc''' is the WOW-weighting vector of (L, acc'') and Q, $acc''(a) = acc'(a)/\sum_{b\in L} acc'(b)$, $acc'(a) = \min(\max_{b\leq a} acc(b), \max_{b\geq a} acc(b))$, $acc(a) = \sum_{f(x_j)=a} p(x_j)$ and where b is the element next to b in L. This is, $b = \max\{x | x \in L, x < a\}$.

Where the WOW-weighting vector is computed according to the following definition:

Definition 2: Let $(a_i, p_i)_{i=1,N}$ be a pair defined by a value a_i and the importance of this value (the value p_i) expressed in a given domain $D \subset \mathbb{R}^+$, and let Q be a fuzzy non-decreasing fuzzy quantifier. Then, the WOW-weighting vector $\omega = (\omega_1, \cdots, \omega_N)$ for $(a, \mathbf{p}) = ((a_1, \cdots, a_N), (p_1, \cdots, p_N))$ and Q is defined as follows:

$$\omega_i = Q\Big(\frac{\sum_{j\leq i} p_{\sigma(j)}}{\sum_{j\leq N} p_{\sigma(j)}}\Big) - Q\Big(\frac{\sum_{j<i} p_{\sigma(j)}}{\sum_{j\leq N} p_{\sigma(j)}}\Big)$$

where $\{\sigma(1), ..., \sigma(N)\}$ is a permutation of $\{1, ..., N\}$ such that $a_{\sigma(i-1)} \geq a_{\sigma(i)}$ for all $i = \{2, ..., N\}$ (i.e. $a_{\sigma(i)}$ is the i-th largest element in the collection $a_1, ..., a_N$), and Q is a non-decreasing fuzzy quantifier. This is, Q is a monotonic function (i.e., $Q(a) \geq Q(b)$ for all $a > b$) such that $Q(0) = 0$ and $Q(1) = 1$.

III. Defuzzification Procedures

In this section we review some of the defuzzification procedures. For more details see [12] and [10].

Defuzzification methods have been studied from different perspectives. Yager [10] views defuzzification in the more general framework of a *selection problem.* This is, selecting an element using the information represented in the fuzzy set. Additional knowledge can also be taken into account in this selection (e.g. using constraints about values of variables in [4]). This latter case corresponds, according to Yager [11], to the so-called *Knowledge based defuzzification.*

To study the defuzzification process, Yager proposed a general architecture. This is shown in Figure 1. Following this figure, the output of the fuzzy rule based system is a fuzzy set F (we will use μ_F to denote its membership function). The defuzzification of this fuzzy set is achieved via a two stage process. First, the fuzzy set F is transformed into a probability distribution P, and then one element is selected from the probability distribution (two general methods $S1$ and $S2$ – see below – are considered for selection). In [11], Yager introduces an additional stage for transforming the original fuzzy set into a *transformed* one F' (with membership function $\mu_{F'}$) using additional knowledge.

Here we consider an hybrid approach consisting on only two stages as in Figure 1 but in which the first stage (this is called *fuzzy set transformation*) considers all the additional knowledge, and its output F' can be another fuzzy set. The second stage is named *Element selection.* This is shown in Figure 2.

We now review some of the existing alternative approaches for the two stages of fuzzy set transformation and element selection. The following two subsections are devoted to these aspects.

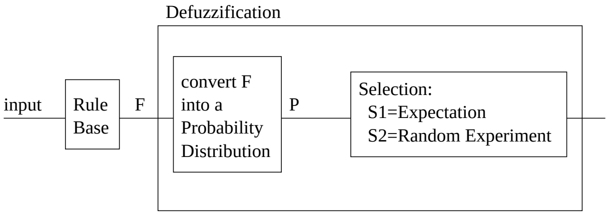

Fig. 1. Defuzzification process following Yager

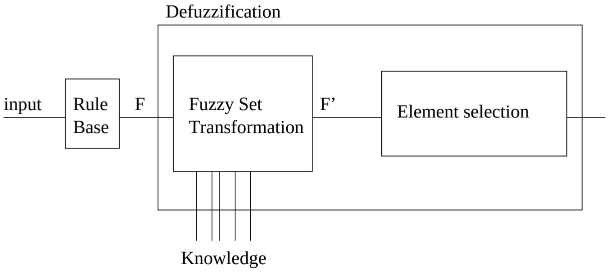

Fig. 2. Our approach to the defuzzification process

A. Fuzzy Set Transformation

Below we list some of the appropriate fuzzy set transformation processes. An alternative method based on clustering is described in [11].

Normalization: Membership values are scaled so that they add to one. This is:

$$\mu_{F'}(x) = \frac{\mu_F(x)}{\sum_{x_i\in X} \mu_F(x_i)}$$

This process can also be understood as transforming the fuzzy set into a probability distribution. The well-known Center of Area method (see e.g. [12], [10] for its description) requires the application of this transformation.

Selection of most possible values: Objects x_i in X with a larger membership value are selected. Let $F_{max} = max \mu_F(x)$, then:

$$\mu_{F'}(x) = \begin{cases} 1 & \text{if } \mu_F(x) = F_{max} \\ 0 & \text{otherwise} \end{cases}$$

The Mean of Maxima method requires the application of this transformation.

α-cut of the membership function: All values with a membership value less than a given α are disregarded.

$$\mu_{F'}(x) = \begin{cases} \mu_F(x) & \text{if } \mu_F(x) \geq \alpha \\ 0 & \text{otherwise} \end{cases}$$

This transformation is applied to avoid the inclusion of values that have a possibility below a given threshold.

As these definitions are functionally defined, composition of transformations are possible in a single defuzzification method. For example, we can apply a given α-cut and then normalize the resulting membership function to obtain a probability distribution that disregards some values with a low possibility.

B. Element Selection

For element selection, Yager [11] made the classification given below. The two classes considered roughly correspond to the two approaches for defuzzification informally reviewed in the introduction (and to $S1$ and $S2$ in Figure 1): element selection and aggregation.

Blending methods: The defuzzified value is obtained as the combination of available solutions. These methods usually use some kind of average to combine the solutions. Differences between methods correspond to different ways of averaging the values. For example, the following aggregation operators have been considered in the literature:

1. **Arithmetic mean of the values:** The Mean of Maxima can be computed using this procedure for element selection.
2. **Weighted mean of the values:** Usually weights are linearly proportional to the membership values in μ_F. This would be the case of the Center of Area. The Mean of Maxima can also be computed using this approach (in this case, all weights are equal).

Celibate methods: They do not combine solutions but take one of the elements of X as its solution. Some particular examples of celibate methods are:

1. Random selection of one of the elements (with random numbers following e.g. a Normal distribution).
2. Random selection with the probability of selecting an element being proportional to its fuzzy membership value. This is the RAGE (RAndom GEneration defuzzification) family of methods [11].

IV. Aggregation and Defuzzification Procedures

As seen in Section III, aggregation is one of the blending methods for element selection. In particular, given a membership function μ_F on the (discrete) reference set X, the defuzzification of μ_F can be seen as the aggregation of the elements $x_i \in X$ with respect to the *weighting vector* μ_F. Thus, $\mu_F(x_i)$ is interpreted as the weight of x_i.

According to this, for a given measure scale (e.g., numerical, ordinal, nominal), all aggregation operators in that kind of scale that use as additional information a numerical weighting vector can be used for defuzzification.

The other relationship between aggregation and defuzzification procedures is that the latter can be seen as an aggregation procedure when values are weighted. This is, given a set of values $a_1, \cdots, a_N$ (with $a_i \in X$) to be aggregated with weights $p_1, \cdots p_N$ (p_i is the weight attached to a_i), the following *"fuzzy set"* on X can be defined: $\mu_{A,p}(x_i) = \sum_{a_j = x_i} p_j$. Then, we can define the aggregation of $a_1, \cdots, a_N$ with respect to $p_1, \cdots, p_N$ as the defuzzified value of $\mu_{A,p}$.

V. Extending Defuzzification Methods in Ordinal Scales

The architecture described above for defuzzification is suitable for any kind of scale. However, while most membership transformations (as the ones described in this work) are independent of the type of scale, this is not as clear for selection methods. Note that blending methods are difficult to apply due to the difficulty of defining aggregation functions in ordinal scales. In particular, the arithmetic mean and the weighted mean (enumerated above) are not applicable because the values to be aggregated belong to an ordinal scale.

To replace these numerical aggregation operators, the operators described in Section II can be used.

Without considering their properties, the (weighted) plurality rule, the (weighted) median, the weighted minimum and maximum and (after some adaptation) the ordinal weighted mean can be applied. Instead, the Sugeno integral cannot. The main difficulty for the Sugeno integral is that when applied in ordinal scales it needs a fuzzy measure in the same ordinal scale than the values. This is difficult to be defined and, moreover, the only available information is numerical (the weights p_i) instead of ordinal. The second inconvenient us that the role of the fuzzy measure is similar to the role of the weights, and both elements are difficult to be combined.

Of these available and applicable operators, the most appropriate one is the Median. The ordinal weighted mean could be used, but it requires a definition of the t-norms and t-conorms involved in the process. The inconvenience of the plurality rule and the median is that they do not allow for compensation. However, the plurality rule has an additional inconvenient because a small variation on the input data can provoke a large variation in the output. Therefore, the result is not much stable. This is a drawback specially relevant when considering the application of procedures to defuzzification.

The CWOW-Median can also be applied and has the advantages of the Median and, moreover, it allows for some compensation (recall that the median lacks this property) and it also includes a parameterization by means of the fuzzy non-decreasing quantifier Q. As shown in [6], this parameterization allows a smooth transition between the smallest value being aggregated and the largest one. This is achieved with $Q(x) = x^\alpha$ and, respectively, with $\alpha = 0$ and $\alpha = \infty$

The CWOW-Median can be decomposed into two stages to adapt it to the architecture for defuzzification described in Section III. The decomposition of the operator into three components (convex transformation, WOW transformation and median) suggests new defuzzification methods. They are built through the combination of these transformations with alternative (other than Median) selection procedures.

In the next section, we introduce some new fuzzy set transformation functions. Besides of the new transformations based on CWOW-Median, we introduce aggregation based ones and we present an example using the Choquet integral.

A. Fuzzy Set Transformation

We start with the two transformations corresponding to the components of the CWOW-Median operators.

Convex membership function: This is to permit the selection of a value with a null membership function if it is located between elements with non-null membership functions. This is solved making the fuzzy set a convex fuzzy set. Given the fuzzy set μ_F, the convex fuzzy set $\mu_{F'}$ is defined as:

$$\mu_{F'}(x) = \min(\max_{b \leq x} \mu_F(b), \max_{b \geq x} \mu_F(b))$$

WOW transformation: Given a non-decreasing fuzzy quantifier Q, the new membership function is defined as:

$$\mu_{F'}(l_i) = Q(\frac{\sum_{j \leq i} \mu_F(l_j)}{\sum_{l_j \in L} \mu_F(l_j)}) - Q(\frac{\sum_{j < i} \mu_F(l_j)}{\sum_{l_j \in L} \mu_F(l_j)})$$

Aggregation transformation: Given a membership function, the new membership function is defined using an aggregation operator. This is, the value for a given category is the aggregation of the membership of nearby categories. This is, given an aggregation operator $\mathbb{C}$, we compute $\mu_{F'}(l_i)$ by:

$$\mu_{F'}(l_i) = \mathbb{C}(\mu_F(l_1), \cdots, \mu_F(l_R))$$

The aggregation operator needs to be customized (via a parameterization) for all categories l_i. This can be better expressed by:

$$\mu_{F'}(l_i) = \mathbb{C}_i(\mu_F(l_1), \cdots, \mu_F(l_R))$$

where $\mathbb{C}_i$ means that the operator (or some *internal* parameters) depends on the label l_i.

Note that if the aggregation operator is not customized for each category, the result would be the same for all categories, because in our definition the function always receives the same parameters.

The transformation based on aggregation can be used to make the membership function smoother. This transformation is similar to the smoothing of data in signal processing. An example of using aggregation for fuzzy set transformation is described in subsection V-A.1.

A.1 Example: Choquet Integral for Defuzzification

As said above, the approach for using aggregation operators for fuzzy set transformation is to consider the new value $\mu_{F'}(l_i)$ for a given category l_i as the aggregation of the value $\mu_F(l_i)$ with the values $\mu_F(l_j)$ for $j \neq i$. To compute this value, we use the Choquet integral. The Choquet integral is a numerical aggregation operator that aggregates some values with respect to a fuzzy measure.

Definition 3: A fuzzy measure μ on a set X is a set function $\mu : \wp(X) \rightarrow [0,1]$ satisfying the following axioms:
(i) $\mu(\emptyset) = 0$, $\mu(X) = 1$ (boundary conditions)
(ii) $A \subseteq B$ implies $\mu(A) \leq \mu(B)$ (monotonicity)

Fuzzy measures replace the axiom of additivity in probability measures ($\mu(A \cup B) = \mu(A) + \mu(B)$ when $A \cap B = \emptyset$) by a more general one: monotonicity. Thus, probability measures are also fuzzy measures. Fuzzy measures are used in Choquet integrals to express the importance of a set of information sources and their redundance and complementariness. When additivity is not satisfied, it means that the importance of a set is not the addition of the importance of the elements by themselves. In our case, as we aggregate the membership values of the categories, the measure corresponds to the importance of the category and their relationship with other categories. This can express whether a category can be aggregated with another one or not.

The definition of the Choquet integral follows:

Definition 4: Let μ be a fuzzy measure on X. The *Choquet integral* of a function $f : X \rightarrow \mathbb{R}$ with respect to μ is defined by:

$$(C)\int f d\mu = \sum_{i=1}^{n} (f(x_{s(i)}) - f(x_{s(i-1)}))\mu(A_{s(i)})$$

where $f(x_{s(i)})$ indicates that the indices have been permuted so that $0 \leq f(x_{s(1)}) \leq ... \leq f(x_{s(N)}) \leq 1$, $A_{s(i)} = \{x_{x(i)}, ..., x_{s(N)}\}$ and $f(x_{s(0)}) = 0$.

In our case, and according to the remark in the previous section that the aggregation operator has to be parameterized in a convenient way for each category, we adapt the Choquet integral to compute the membership of the category l_i (this is, to compute the value $\mu_{F'}(l_i)$) as follows:

Definition 5: Let μ_{l_i} be a fuzzy measure on X for category l_i. The *Choquet integral* of the membership function $\mu_F : X \rightarrow \mathbb{R}$ with respect to μ_{l_i} is defined by:

$$\mu_{F'}(l_i) = \sum_{j=1}^{n} (\mu_F(l_{s(j)}) - \mu_F(l_{s(j-1)}))\mu_{l_i}(A_{s(j)})$$

where $f(x_{s(j)})$ indicates that the indices have been permuted so that $0 \leq f(x_{s(1)}) \leq ... \leq f(x_{s(N)}) \leq 1$, $A_{s(j)} = \{x_{s(j)}, ..., x_{s(N)}\}$ and $f(x_{\sigma(0)}) = \emptyset$.

To make the integral applicable, we need a fuzzy measure. This measure depends on our prior knowledge about the categories, and, as said, is different for each category l_i (we denote this measure by μ_{l_i} as above). In our case, we propose the following fuzzy measure (the measure is defined in terms of its Möbius transformation m_{l_i}):

Given a category $l_i \in X$ and a constant $K \in [0,1]$, we define the Möbius transform m_{l_i} of the fuzzy measure μ_{l_i} as follows:

$$m_{l_i}(A) = \begin{cases} K & \text{if } A = l_i \\ K/|A| & \text{if } l_i \in A \text{ and } A \text{ is convex} \\ 0 & \text{otherwise} \end{cases}$$

with a convex set A, we mean a set such that for all a, b in A, all the values c such that $a \leq c \leq b$ also belong to A.

The rational of this measure is that the only sets A to be taken into account when computing the membership value of the category l_i are the ones that are *connected* with l_i. This is, if a set contains l_j with $l_j > l_i$ but does not contains a category in between, the set is not considered. The relative importance of a set is larger for smaller sets, and the maximum importance is given to the set with the category l_i.

To avoid that $\mu_{F'}(l_i) > 1$, we need that the fuzzy measure defined above is normalized in the sense that $\mu_{l_i}(X) = 1$. This can be achieved in two ways. The simplest way is to define an alternative Möbius transform as follows:

$$m'_{l_i}(B) = m_{l_i}(B) / \sum_{A \subseteq X} m_{l_i}(A)$$

and use this function to compute the measure μ_{l_i}. Recall (see [13]) that this is achieved as follows:

$$\mu_{l_i}(B) = \sum_{A \subseteq B} m'_{l_i}(A)$$

An alternative way is to use the same normalization factor for all fuzzy measures μ_{l_i}. This means to define the normalization factor (NF) as follows:

$$NF = \max_{l_i \in X} \sum_{A \subseteq X} m'_{l_i}(A)$$

and define

$$m''_{l_i}(B) = m_{l_i}(B)/NF$$

The first normalization approach means that when $\mu_F(x) = \alpha$ for all $x \in X$, then $\mu_{F'}(x) = \alpha$ for all $x \in X$. Thus, all elements are equally possible after transformation.

The second approach is more appropriate when in a constant membership function (as above, $\mu_F(x) = \alpha$ for all $x \in X$) we are more interested in central elements of X. In other words, this means that a given x_i not only receives support from the x_i itself but also from contiguous values x_{i+1} and x_{i-1}. In this sense, extreme values have less support because they have only one neighbor. The same comment applies to extreme values of convex membership functions.

VI. Extending Aggregation Operators in Ordinal Scales

The new defuzzification processes introduced in Section V define, when combined with the selection processes in Section III, some new defuzzification procedures. These procedures are suitable, according to the transformation found in Section IV, for aggregation. In particular, the use of RAGE selection introduces several new aggregation operators.

Some examples of these new operators are the following:

- The combination of convex membership function and the RAGE selection to define the Convex-RAGE defuzzification process.
- The combination of the convex membership function, the WOW transformation and the convex RAGE leads to the CWOW-RAGE defuzzification method.
- The combination of convex membership function, the WOW transformation and the median leads to the CWOW-Median.
- The Choquet transformation (aggregation with fuzzy measure) and the median would lead to the Choquet median.

VII. Conclusions

In this work we have considered defuzzification methods based on the architecture described by Yager in [11]. We have reviewed transformation functions and methods for element selection. The combination of one element of each stage leads to several defuzzification methods. All these methods can also be seen as aggregation procedures. In particular, the use of random selection procedures in combination with membership function transformation procedures lead to new aggregation operators.

VIII. Acknowledgments

Partial support of the European Community under the contract "CASC" IST-2000-25069 and of the CICYT under the project "STREAMOBILE" (TIC2001-0633-C03-01/02) is acknowledged.

References

[1] G. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall, U.K., 1995.

[2] V. Torra, A review on the construction of Hierarchical fuzzy systems, Int. J. of Intel. Systems, vol. 17 pp. 531-543, 2002.

[3] L. A. Zadeh, Fuzzy sets, Information and control, vol. 8 pp. 338-353, 1965.

[4] V. Torra, A. Sodan, A Multi-Stage System in Compilation Environments, Fuzzy Sets and Systems vol. 105 pp. 49-61, 1999.

[5] V. Torra, Aggregation of linguistic labels when semantics is based on antonyms, Int. J. of Intel. Syst., vol. 16 pp. 513-524, 2001.

[6] J. Domingo-Ferrer, V. Torra, Median based aggregation operators for prototype construction in ordinal scales, submitted.

[7] F. Herrera, E. Herrera-Viedma, J.L. Verdegay, A Sequential Selection process in Group Decision Making with a Linguistic Assessment Approach, Information Science, vol. 85 pp. 223-239, 1995.

[8] V. Torra, The Weighted OWA operator, International Journal of Intelligent Systems, vol. 12 pp. 153-166, 1997.

[9] G. Mayor, J. Torrens, On a Class of Operators for Expert Systems, Int. J. of Intelligent Systems, vol. 8, pp. 771-778, 1993.

[10] R. R. Yager, D. P. Filev, Essentials of fuzzy modeling and control, John Wiley & Sons, New York, 1994.

[11] R. R. Yager, Knowledge-based defuzzification, Fuzzy Sets and Systems, vol. 80 pp. 177-185, 1996.

[12] D. Driankov, H. Hellendoorn, M. Reinfrank, An introduction to Fuzzy Control, Springer-Verlag, USA, 1996.

[13] J.-L. Marichal, Aggregation operators for multicriteria decision aid, PhD Dissertation, Faculte des Sciences, Universite de Liege, Liege, Belgium, 1999.

OWA operators in data modeling and re-identification

Vicenç Torra

Abstract— **This paper is devoted to the application of aggregation operators and to the application of OWA operators to data mining. In particular, we consider two application of OWA operators in this field: model building and information extraction. The latter application is oriented to the re-identification procedures.**

Keywords— **Aggregation Operators, Weighted Mean, OWA Operators, Learning Models, Re-identification Methods, Record Linkage, privacy preserving data mining.**

I. Introduction

Information fusion techniques, in general, and aggregation operators, in particular, are currently being used in several scientific fields. In fact, their use is rapidly increasing because, on the one hand, data is gradually obtained in an easier way and, on the other hand, computational power has largely increased so that systems that combine information from several experts or sensors are nowadays feasible. Even more, it is also possible to compute several solutions using different approaches and then combine the alternative solutions. This is the case of bagging, boosting and related approaches used for building data models in machine learning. In these approaches, several models are built using different mechanisms and, then, a decision making module (e.g. voting [16], [2], the weighted mean [17] or the OWA operator [42]) is put on top of the models.

In general, in the field of artificial intelligence, data fusion techniques are mainly used for two main purposes (i) when a system has to make a decision, or (ii) when it needs a comprehensive representation of its domain.

In the first case, an alternative has to be selected or built from several ones. The typical case for selection is to consider several criteria for each alternative (this corresponds to a multi-criteria decision-making problem) and the best alternative is usually chosen [27] in a two phase process: (i) the aggregation of the degree of satisfaction for all criteria, per decision alternative; and (ii) the ranking of the alternatives with respect to the global aggregated degree of satisfaction. Instead, when the alternative has to be built from the existing ones, fusion corresponds to the whole building process and it has to consider the importance and reliability of the alternatives and of the approaches used to build these alternatives. Plan merging can be seen from this point of view.

In the second case, a system builds the representation of its environment from some background knowledge embedded in the system and some knowledge supplied by some information sources (e.g. experts or sensors). Naturally, the knowledge has to be "reliable" and extend on the whole domain of system's actuation. However, the information supplied by a single information source is often not reliable enough and also too narrow in relation to the working domain. In this case, the information provided from several sensors or experts are combined to improve data reliability and accuracy and to include some features that are impossible to be perceived from individual sensors. Note that information fusion for knowledge representation can be either applied at the time of defining the background knowledge (e.g. using several experts' knowledge) or at run-time (either combining different pieces of new information or combining some new information with some knowledge already established in the system).

Nevertheless, other uses of data fusion techniques are conceivable in the artificial intelligence field. In particular, fusion is useful in data mining and knowledge discovery for two types of applications. On the one hand, aggregation operators and, in general, any data fusion model, are suitable for building data models. This capability relies on a result published in [34]. In this work, it is proved that a model defined in terms of a hierarchy of quasi-weighted means (a kind of aggregation operator – see [12] or [39] for details on the operator) is a universal approximator and, as such, it can be used for representing any arbitrary complex function. Therefore, given a data set it is possible to build a model of this data set using aggregation operators. On the other hand, aggregation operators can be used to extract useful information from raw data and, thereafter, other programs can use the structures that otherwise would remain implicit.

In this work we consider both approaches: the use of aggregation operators to build data models, and its use to extract implicit information. In the former case, due to the large number of aggregation operators we concentrate in some of them for which learning models are simple (small cost) and, thus, applicable to data mining. In particular, we focus on the OWA operator [47]. In relation to the latter case, we show an application to the re-identification of individuals in data files with non-common variables. We focus on the use of aggregation operators to extract implicit information from files and their application to re-identification.

The structure of this paper is as follows. In Section II, we review the state of the art of the two approaches. Then, in Section III we review some definitions that are needed later on in this work. Section IV is devoted to describe a method to determine models for quasi-weighted means. Section V corresponds to the use of aggregation operators to extract information and its application to re-identification. The

V. Torra is with the Institut d'Investigació en Intel·ligència Artificial - CSIC, Campus UAB s/n, 08193 Bellaterra, Catalunya, Spain. E-mail: vtorra@iiia.csic.es

work finishes in Section VI with some conclusions.

II. State of the art

In this Section we first review recent work on the use of aggregation operators for building models. Then, we give an overview of re-identification algorithms.

Aggregation operators have been widely studied. See, *e.g.*, [4], [19], [40] for a detailed state-of-the-art description of the field. See [46] for a recent overview on aggregation operators.

A. Aggregation operators for building data models

Theorem 1 in [34] establishes that hierarchies of quasi-weighted means are universal approximators. This result implies that quasi weighted means in particular, and aggregation operators in general, are suitable tools for modeling data and, as a consequence, methods can be developed to build models for complex data. When efficient tools are developed, methods can be used for large data bases.

At present, methods to determine the suitable aggregation model have been focused on mechanisms to determine the parameters of the operator once the aggregation operator is selected. Several approaches have been developed in the last years. They can be broadly classified in two groups.

On the one hand, there are some methods based on the assumption that there exists an expert that supplies crucial information that is used later on to extract the parameters of the selected aggregation function. This is the case of Saaty's Analytical Hierarchy [28] process (used to determine the weighting vector of a weighted mean). A similar approach is the one followed by O'Hagan in [22] to determine the weighting vector of the Ordered Weighting Averaging (OWA) operator. He requires that a user supplies the so-called *orness*, a measure of how much large values influence the outcome of the operator. This method was further developed in [5].

On the other hand, there are some methods that do not require the presence of an expert but the existence of a set of examples. In this case, an example consists on the input values (i.e., values to be aggregated) and the expected result (i.e., the value that the model estimates). From these examples, the parameters of the operators are inferred and the model is fitted on the data. This approach is rooted on estimation theory where two random variables X, Y are considered such that Y is said to depend on X when the distribution $Y|X$ is different from the distribution Y. This is, the distribution posterior to observation is different from the a priori one. Then, a model is built to estimate the value of Y such that the variance of error is minimized. See [13], [3] for a more detailed discussion of learning parameters from this perspective.

Several works can be found in the literature about parameter learning from examples. For example, [9], [10] studies the determination of the weighting vector for the OWA operator. In a similar way, [33] studies the learning for both the weighted mean and the OWA operator. [12] and [31] deal with the learning of fuzzy measures for the Choquet integral and [13] compares different approaches for modeling using the Choquet integral.

In the framework of data mining, the second approach is of interest. It allows to build a data model from existing data. However, not all the approaches to build the model are suitable as some methods have a high computational cost. Iterative computation with a cost proportional (linear or even higher) to the number of examples is prohibitive. For example, genetic algorithms based approaches [20] are, usually, extremely costly as the fitness function needs to compute at each step an aggregated value for all the examples. This is also the case for the method described in [9], [10] based on gradient descent. However, in this latter case, if it is possible to start with a "quite good solution", then it can be revised as long as time and resources are available. A good alternative for building models in the case of quasi-weighted means is to use active set methods. These methods are iterative ones and thus can obtain different solutions according to the "available" time. In this approach, the cost of the initial step is proportional to the number of examples, then, once the iterative process is started, the cost of a single step is proportional to the number of considered variables. Moreover, the number of steps of the iterative process is bounded and thus the best solution can be found in a finite time.

B. Re-identification of individuals.

Re-identification happens when two entities are detected as corresponding to the same object or individual. For example, when some sensitive and confidential data is linked to a particular individual. Record linkage is one of the most general re-identification method. Its goal [21], [26] is to link records in separate files that relate to the same individual or household. These methods were developed to improve the quality of the data and are nowadays used in data cleaning [24] for distributed and non-homogeneous databases. Such databases typically [41] contain information about the same individuals described using the same variables that, frequently, do not match due to accidental distortion of the data. Record linkage is applied in such cases to find the records that correspond to the same individuals and to make databases consistent. Existing tools for this purpose (e.g. Integrity [14]) use statistical and Artificial Intelligence techniques to determine matching between records. Multi-database mining, that intends to extract knowledge from non-homogeneous databases (see e.g. [50]), also benefits from these tools.

[44], [11] and [41] describe the main approaches for record-linkage. The usual case is to consider files that share a set of variables. In this case, the main difficulty is that a matching procedure among pairs of records is usually not enough to link the records. This is so because data files are subject to errors (either due to intentional or to accidental distortion) and thus not only equal values should match but also similar ones. As [45] points out, "the normal situation in record linkage is that identifiers in pairs of records that are truly matches disagree by small or large amount and that different combinations of the non-unique,

error-filled identifiers need to be used in correctly matching different pairs of records".

This situation of files sharing a set of variables is usually dealt by probabilistic record linkage [44] or distance-based record linkage [23]. The former is based on estimating (using the EM algorithm – see [6] and [15] for details) conditional probabilities of coincidence of the values of a particular variable when a true match or a true non-match is assured. This is, which is the probability that the corresponding values for a particular variable is the same, when two records are known to correspond (or not correspond) to the same individual. Then, given a pair of records the pair is classified as either corresponding to the same individual or not according to an index computed from these probabilities. While first methods assumed conditional independence between variables [15], [21], more recent works avoid such assumption. See [41] for a detailed description of probabilistic record linkage and [45] for a description of current approaches and research topics. The distance-based record linkage consists on linking a record with the more similar one. This approach relies on the existence of a distance function. [41] compares both methods and concludes that the probabilistic one is slightly better (re-identifies more records) for categorical variables and that the distance-based one is more appropriate for numerical variables. Recently, alternative methods based on other assumptions have been introduced in the literature. For example, [1] describes a method based on clustering techniques.

Although most methods follow the ideas explained above that files share a set of common variables, other situations are also possible. In particular, it is also of interest the case of files not sharing any variable (or only a few of them). In this case, re-identification is partially possible but being of a different nature because it cannot be based on the comparison of values from records of different files but corresponding to the same variable.

Record linkage for files not-sharing variables is of interest when considering data files with similar information (e.g. economical variables) from consecutive time periods (e.g. two different years) concerning to almost the same individuals (e.g. the companies of a certain region). In this case, although the variables are not the same, "similar" behavior of variables in both files allows for the re-identification of the individuals. Naturally, the more similar the behavior, the better for re-identification. Nevertheless, although this is a subject of increasing interest, no much effort has been devoted to the subject.

In [35], we gave a first approximation to the re-identification for files not sharing variables. A more exhaustive analysis of the approach using real data is described in [8]. In [35] and [8], some basic guidelines are established to allow for re-identification: (i) files share a set of individuals and (ii) some relationships between individuals are kept across files. In [35] and [8] these relationships are established using clustering algorithms. These methods differ from [1], that also uses clustering for re-identification, on the way clusters are defined and how the records are linked once their corresponding clusters are known.

Here we follow, a different approach suitable for quantitative variables. We show that some aggregation operators (in particular, we focus on the Ordered Weighted Averaging - OWA - [47] operators) are a suitable way to extract implicit structures from data. This work extends [37]. Here, we give empirical results and we proof that the number of re-identified elements are significant.

III. Preliminaries

In this section we review some aggregation operators that are used latter on in this work. In particular, definitions for the weighted mean, quasi-weighted mean and the OWA operator (the one based on a weighting vector and the one based on non-decreasing fuzzy quantifiers) are given.

Definition 1: A vector $w = (w_1 \cdots w_n)$ is a *weighting vector* of dimension n if and only if $w_i \in [0,1]$ and $\sum_i w_i = 1$.

Definition 2: Let w be a weighting vector of dimension n, then a mapping $WM_w : \mathbb{R}^n \to \mathbb{R}$ is a *Weighted Mean (WM)* of dimension n if

$$WM_w(a_1, \cdots, a_n) = \sum_i w_i a_i$$

Definition 3: Let w be a weighting vector of dimension n, let f be a strictly increasing function, then a mapping $QWM_w : \mathbb{R}^n \to \mathbb{R}$ is a *Quasi-weighted Mean (QWM)* of dimension n if

$$WM_w(a_1, \cdots, a_n) = f^{-1}(\sum_i w_i f(a_i))$$

For properly selected functions $f(x)$, quasi-weighted means generalize some well known aggregation operators. See, for examples, that with $f(x) = Kx + K'$ we obtain the weighted mean and with $f(x) = Klogx + K'$ we obtain the geometric mean.

Definition 4: [47]. Let w be a weighting vector of dimension n, then a mapping $OWA_w : \mathbb{R}^n \to \mathbb{R}$ is an *Ordered Weighted Averaging (OWA) operator* of dimension n if

$$OWA_w(a_1, \cdots, a_n) = \sum_i w_i a_{\sigma(i)}$$

where $\{\sigma(1), \cdots, \sigma(n)\}$ is a permutation of $\{1, \cdots, n\}$ such that $a_{\sigma(i-1)} \geq a_{\sigma(i)}$ for all $i = 2, \cdots, n$. (i.e., $a_{\sigma(i)}$ is the i-th largest element in the collection $a_1, ..., a_n$).

This definition of the OWA operator requires a weighting vector of fixed dimension being its dimension the number of elements to aggregate. An alternative definition based on decreasing fuzzy quantifiers exists that can be used for data vectors of arbitrary size. This alternative definition allows the comparison of different size data vectors: comparison with respect to the outcome of the OWA operator. Similarity of two data vectors can then be measured as a function of the differences between the outcomes of the OWA operators applied to the vectors.

Definition 5: [48]. A function $Q : [0,1] \to [0,1]$ is a *non-decreasing fuzzy quantifier* if $Q(0) = 0$, $Q(1) = 1$ and for all x, y in $[0,1]$, $x < y$ implies $Q(x) \leq Q(y)$.

Definition 6: [48], [49]. A mapping $OWA_Q : \mathbb{R}^n \rightarrow \mathbb{R}$ is an *Ordered Weighted Averaging (OWA) operator* of dimension n if

$$OWA_Q(a_1, \cdots, a_n) = \sum_i w_i a_{\sigma(i)}$$

where $w_i = Q(i/n) - Q((i-1)/n)$ and where σ is defined as in Definition 4.

In this way, an OWA operator with a suitable fuzzy quantifier can compute a value with which *all* information sources agree, *at least one* source agree, *about half* agree, etc. Note that the same quantifier allows the computation of the aggregated value for an arbitrary number of inputs (sources) and that the quantifier only refers to a proportion of the sources but not to an exact number.

Therefore, the OWA operator, specially when defined in terms of a fuzzy quantifier, is a flexible tool that is appropriate when the number of variables is not known or can change in different instantiations of the same problem. This is the case in the re-identification problem described in Section V and similar situations can be envisioned in ensemble methods (new partial models are included in the system without modifying the decision making module). In fact, the flexibility of OWA operators compare positively in relation to other operators as e.g. the weighted mean (where the number of sources has to be fixed before hand and weights are assigned to sources). Moreover, more complex operators like the Choquet and Sugeno integrals present additional difficulties because the number of parameters tend to be extremely large (2^n where n is the number of sources or variables).

IV. Aggregation operators for building data models.

This section is focused on mechanisms for learning parameters for aggregation operators from examples. Examples are assumed to be in accordance with Table I. This is, it consists on M different examples, each of them consisting on the values supplied by N information sources and the *correct* outcome that we intend to estimate from these values. Therefore, each example consists on $N + 1$ values being $(a_1^i a_2^i \ldots a_N^i | b^i)$ the ones for the i-th example where a_j^i is the value supplied by the j-th information source and being b^i the ideal outcome for the same example.

a_1^1	a_2^1	$\ldots$	a_N^1	|	b^1
a_1^2	a_2^2	$\ldots$	a_N^2	|	b^2
$\vdots$	$\vdots$		$\vdots$		$\vdots$
a_1^M	a_2^M	$\ldots$	a_N^M	|	b^M

TABLE I
Data examples.

Given a set of examples and assuming that the function used to aggregate $\mathbb{C}$ is known, the goal is to determine the parameters of $\mathbb{C}$. When $\mathbb{C}$ is the weighted mean, this is to find the weighting vector w so that error is minimized. Error is measured for each example in terms of the difference between the ideal outcome (i.e., b^j) and the real outcome. This is, in the general case:

$$(\mathbb{C}(a_1^j, \ldots, a_n^j) - b^j)^2$$

and when $\mathbb{C}$ is the weighted mean:

$$(WM_{\mathbf{w}}(a_1^j, \ldots, a_n^j) - b^j)^2$$

Therefore, following the notation in Table I, the function to minimize is:

$$D_{\mathbb{C}}(parameters(\mathbb{C})) = \sum_{j=1}^{M} (\mathbb{C}(a_1^j, \ldots, a_n^j) - b^j)^2 \qquad (1)$$

However, this problem is usually a constrained one because there usually exist constraints over the parameters. This is the case of the weighted mean, where w is a weighting vector. In this case, the problem can be formalized in the following way:

$$\begin{array}{l} Minimize\ D_{WM}(w) \\ Subject\ to \\ \qquad \sum_{i=1}^{N} w_i = 1 \\ \qquad w_i \geq 0 \end{array}$$

where $D_{WM}(w) = \sum_{j=1}^{M} (WM_w(a_1^j, \ldots, a_n^j) - b^j)^2$.

A. Solving the optimization problem

The problem formulated above is a typical optimization problem where a function has to be minimized subject to a set of constraints. There exist several methods to solve these problems according to the function to minimize (quadratic, convex, ...) and the type of restrictions that apply (linear constraints, equality constraints, ...). When the distance to minimize is either $D_{WM} = \sum_{j=1}^{M} (WM_{\mathbf{w}}(a_1^j, \ldots, a_n^j) - b^j)^2$ or $D_{OWA} = \sum_{j=1}^{M} (OWA_{\mathbf{w}}(a_1^j, \ldots, a_n^j) - b^j)^2$ the problem to solve is a quadratic one subject to linear constraints. In such a case several methods apply. For example, [9], [10] and [33] applied two different approaches: [9], [10] used the gradient descent for the OWA operator and [33] used active set methods for the weighted mean and the OWA operator.

In order to avoid the inconvenience of dealing with the inequality and equality constraints ($w_i \geq 0$ and $\sum_i w_i = 1$) when applying gradient descent to the OWA operator, [9], [10] reformulated the problem. Instead of considering the learning of the weighting vector w, they considered the learning of a vector $\mathbf{\Lambda} = (\mathbf{\lambda_1} \ldots \mathbf{\lambda_N})$ from which weights were extracted as follows:

$$w = (e^{\lambda_1} / \sum_{j=1}^{N} e^{\lambda_j} \ \ldots \ e^{\lambda_N} / \sum_{j=1}^{N} e^{\lambda_j})$$

In this way, any vector $\mathbf{\Lambda} \in \mathbb{R}^{\mathbf{N}}$ leads to a weighting vector. Therefore, the problem of learning weights for the OWA operator (we assume again that available data follows the description in Table III) is equivalent to the minimization of:

$$D_{OWA}(p) = \sum_{j=1}^{M} OWA_w(a_1^j, \ldots, a_n^j) - b^j)^2 \qquad (2)$$

where $\mathbf{w} = (\mathbf{e}^{\lambda_1} / \sum_{\mathbf{j=1}}^{\mathbf{N}} \mathbf{e}^{\lambda_\mathbf{j}} \; \ldots \; \mathbf{e}^{\lambda_\mathbf{N}} / \sum_{\mathbf{j=1}}^{\mathbf{N}} \mathbf{e}^{\lambda_\mathbf{j}})$

[33] presents with great detail an alternative approach based on active set methods. Active set methods rely on the simplicity of computing the solution of quadratic problems with linear equality constraints. Based on this, iterative algorithms have been developed in which at each step inequality constraints are partitioned into two groups: those that are to be treated as active (considered as equality constraints) and inactive (essentially ignored). Once a partition is known, the algorithm proceeds moving on the surface defined by the working set of constraints (the set of active constraints) to an improved point. In this movement some constraints are added to the working set and some others are removed. This process is repeated until the minimum is reached. When the function to minimize is convex (as they are D_{WM} and D_{OWA}) the method finds the minimum and although the method is iterative the final minimum is not influenced by the initial weighting vector.

Gradient descent requires the computation of the gradient at each step; and the computation of the gradient in successive steps require the evaluation of the examples. As the set of examples is usually very large in data mining domains, it is difficult to apply this approach in this field unless an initially good solution is considered and the iterative process is limited to refine the initial solution (this was the case in [38]). Other disadvantages for gradient descent are its slow convergence and the problem of having more than one $\mathbf{\Lambda}$ vector that correspond to the same weighting vector. This, together with the fact of being an iterative process provokes that the final result depends often on the initial weighting vector.

Active set methods require an initial computation of a square matrix from the original data. Although the dimension of this matrix is the number of variables, its construction cost is on the order of $N * N * M$ (being, as before, N the number of variables and M the number of elements) and thus it is a time-consuming task proportional to the number of elements. However, once this matrix is built, each step of the iterative process has a polynomial cost on N.

Thus, for the weighted mean and the OWA operator is more efficient to use the second approach. However, when the function to minimize is not quadratic (as it is the case of the Weighted OWA operator [32]), active set methods are complex and difficult to implement because it is not easy to find the minimal solution at each step and, instead, gradient descent can be used. Software packages can be used at this time.

Both learning methods have been applied to determine the weights for the weighted mean and the OWA operator (in practice, the only difference when learning the weights for the OWA operator in relation to the learning for the weighted mean is that each row in the data matrix has to be reordered according to the permutation σ).

The same approach can be used to learn the weights for a selected quasi-weighted mean (see [39]). This is, a quasi-weighted mean with a known generator function. Let f be the generator of the quasi-weighted mean, then the parameters are determined considering in the distance to minimize the following expression:

$$(\sum_i w_i f(a_i^j)) - f(b^j))^2$$

instead of the original one:

$$(f^{-1}(\sum_i w_i f(a_i^j)) - b^j)^2$$

This is so because both expressions would lead to a similar distance when weights are learned, and the former is easier to minimize. It allows to compute in an initial step the values $f(a_i^j)$ for all i and j, and then the methods described for the weighted mean can be applied. Considering the other expression, the minimization problem becomes a non quadratic problem, and thus more difficult to solve. In a recent work, Beliakov [3] has also considered such an optimization problems.

B. Examples

Example 1: Three examples are considered below. They are taken from the machine learning repository [18]: the *iris* data file (4 variables and 150 examples), the *abalone* data file (8 variables and 4177 examples) and the *ionosphere* data file (34 variables and 351 examples – one, that is always zero, is removed). To use this files some preliminary work was required. First, we had to replace all symbolic variables by numerical ones. These were the changes performed: the classes iris-setosa, iris-versicolor and iris-virginica in the *iris* data file were replaced by numerical values 1.0, 2.0 and 3.0; the three categories M, F, and I (infant) in the variable Sex in the *abalone* data file were also replaced by the numerical values 1.0, 2.0 and 3.0; and the classes "g" and "b" in the *ionosphere* data file were replaced by 1.0 and 0.0. Second, we normalized all the variables in the $[0, 1]$ interval.

For each of the resulting files, active set methods were applied and two models were built: one for the weighted mean and the other for the OWA operator. The corresponding distances are given in Table II.

This example shows the suitability of the approach and its cost make it appropriate for large data files. Note that in the ionosphere case, the number of variables is large (34 variables) but the computational cost in the iterative process only depends on the number of variables.

In [33], this approach was compared against [10] for some toy examples described in this latter work and results showed a better performance of our approach (error was reduced from 0.002156 to 0.001256).

	iris	abalone	ionosphere
D_{WM}	3.1544	37.5189	55.9145
D_{OWA}	6.3197	43.8082	62.9477

TABLE II
Optimal distances for the iris, abalone and ionosphere data files using learned parameters for weighted mean D_{WM} and OWA operator D_{OWA}.

V. Aggregation operators to extract information from data.

It is a well-known fact that an aggregation operator summarizes the information that the information sources supply. In particular, practical use of these operators is motivated by their capability in reducing the uncertainty associated to the data, compensating redundancy and, in general, to extract relevant information. These properties are the ones that make these operators interesting for extracting information from raw data.

Also, these properties are interesting for re-identifying individuals from data files not sharing variables as, a priori, in such situations the only available information is the one available in the files. In this case, if files contain the "same information" a working hypothesis is that when the aggregation is applied to two records corresponding to the same individual, the relevant information emerges from the raw data. According to this, we assume that for each record a representative that is *somehow* independent of the actual data can be computed. This independence has to hold so that the two representatives (one for each file but from the same individual) are similar. Also, ideally, the representatives are similar although variables (and the corresponding values) are different. In fact, this hypothesis holds if variables are correlated (or, more precisely, if one set of variables *as a whole* is "correlated" with respect to the other set of variables). This is, for example, the case of "income" and "size of household". Therefore, two files, one containing the information corresponding to "income" and the other with the information corresponding to "size of household" could be used for re-identification of individuals.

However, as the representative value has to be *somehow* independent of the variables, not all aggregation operators can be applied. In fact, it seems that the best ones are those that are commutative (i.e., $\mathbb{C}(a_1, \cdots, a_n) = \mathbb{C}(a_{\sigma(1)}, \cdots ,_{\sigma(n)})$ for any permutation σ) because particular variables do not have any influence on the output. According to this, the weighted mean and the Weighted OWA [32] are not applicable here. In fact, the OWA operator is the only commutative Choquet integral. An additional element to be taken into account is OWA's flexibility with respect to the number of parameters (this was described in Section III). For all this, we have used OWA operators defined in terms of fuzzy quantifiers. It has to be said that Sugeno integral [30] with commutative fuzzy measures could also be used. However, these integrals are usually applied to data belonging to ordinal scales instead of numerical scales and in this work we limit our approach to the case of numerical data.

According to what has been introduced here, the assumptions listed below direct this work. In the following we assume that we want to link records that belong to two different files.

Hypothesis 1: Both files share a large set of common individuals.

Hypothesis 2: Data in both files contain, implicitly, similar structural information.

Hypothesis 3: Structural information can be expressed by means of numerical representatives for each individual.

Hypothesis 4: Aggregation operators can summarize the information of each individual.

The first hypothesis implies that re-identification is possible, as there are records to be linked because they correspond to the same individual. The second hypothesis is to say that there are similarities between different individuals that are kept *more or less* constant in both files. We call these similarities *structural information.* The third hypothesis and the fourth one are the ones that justify the use of aggregation procedures.

A. Using OWA operators to extract information.

To use OWA operators for information summarization, we need the settlement of OWA operators. First of all, one of the two definitions has to be selected. As previously described, due to the fact that the number of variables can be different in both files, it is appropriate to use the quantifier based definition (i.e., Definition 5). This is so because the same quantifier can be used in combination with any input vector of arbitrary dimension. Using the other approach would require the extension of an n-dimensional weighting vector into a m-dimensional one ($n \neq m$). Although methods exist to do this extension (e.g. the construction of the quantifier in [36]) it is simpler to start with the definition based on the quantifier. Moreover, the use of fuzzy quantifiers allows us to consider families of parameterized quantifiers.

It is a well-known fact that different quantifiers lead to different results of the OWA operator. In fact, different parameterizations correspond to different representatives of the individual. As a priory it is not known which of the representatives is the best one for re-identification, we have considered a set of them. This is, we apply the OWA operator with several parameterizations (we have considered a family of fuzzy quantifiers) obtaining in this way for each file a two dimensional table that follows the one in Table III. For each individual R_i^A in a file A and for each parameterization P_j we have the corresponding aggregated value $c_{i,j}^A$. This is, $c_{i,j}^A$ is the result of applying the OWA operator with the j-th parameterization to the i-th record in file A.

The same process (the same OWA operator with the same quantifiers) is applied to both files. As the number of parameterizations is the same in both files, the same structure is built for both files. Then, usual record linkage techniques can be used to link the new records (now records share the same set of variables). The application of this method and the results obtained are explained in the next section.

A	P_1	P_2		P_t
R_1^A	$c_{1,1}^A$	$c_{2,1}^A$	$\cdots$	$c_{t,1}^A$
$\cdots$	$\cdots$	$\cdots$		$\cdots$
R_n^A	$c_{1,n}^A$	$c_{2,n}^A$	$\cdots$	$c_{t,n}^A$

TABLE III
Summarization structure for file A.

B. Examples.

Example 2: To analyze the feasibility of our approach we have analyzed three artificial problems. These problems have been generated using the publicly available information from the UCL repository [18] we have already used in Section IV-B. This is, the iris, abalone and ionosphere data files. The selection of this data set for structure determination is based on their use of continuous variables and the fact that being public data files experiments can be reproduced.

To use this data for re-identification, two alternatives were possible: re-identification of the examples and re-identification of the variables. In the first alternative, the original file would be split in such a way that all examples but only half of the variables are present in both files. In the second alternative, the original file would be split so that all variables but only half of the examples are present in both files. We have followed the latter approach because it is not sure that half of the variables have enough information about the examples to allow for re-identification. Instead, two randomly chosen subsets of about half of the initial examples (about 175 examples in the case of ionosphere, 2000 examples in the case of abalone, and 75 examples in the case of iris) should give enough information about the structure of the variables. In fact, subsets of these examples are usually used in machine learning [29] because they assume that these subsets have still enough information to model the variable behavior. In other words, the second approach has been used because we assume more redundancy in the examples than in the variables.

To apply the method described above, we have considered an initial normalization step following the usual approach. I.e., Translation of the initial value x in the $[min, max]$ interval into to the value x' in $[0, 1]$:

$$x' = (x - min)/(max - min)$$

After normalization, the file was partitioned into two sets of approximately the same number of records (records were selected at random).

Then, for each variable in each file, the OWA operator has been applied using 10 different parameterizations. Selected quantifiers are: $Q(x) = x^{i/5}$ for $i = 1, \cdots, 10$. In this way, for each of the initial data file we have obtained two files, with the 10 representatives for each variable each. At this point, to re-identify the variables, a record linkage algorithm was applied to each pair of files. We have applied the record linkage algorithm developed by W. Winkler [43] at U. S. Census Bureau (U.S.A.). The number of variables that were correctly re-identified are given in Table IV. In the case of the iris data file, only one link was suggested (but it was incorrect), the other variables were not considered as related.

initial file	re-identified variables	number of variables
iris	0	4
abalone	6	8
ionosphere	10	33

TABLE IV
Number of re-identified variables (in the ionosphere file one of the variables is always zero and it was not considered in the re-identification process).

B.1 Evaluation of the results.

Although the results we have obtained so far are not as good as we would like, and, in the particular case of the ionosphere data file, the number of corrected links is less than a half of the total number of variables, results are quite better than they seem. To evaluate the results we consider the probability of having more than a certain number of correct links, say k, in a random permutation of n individuals. We study below this probability and then we compute the probabilities for the the abalone and ionosphere data files.

Perfect re-identification when two files A and B have the same n individuals correspond to finding a particular permutation π such that for each record i in A, $\pi(i)$ is assigned to each corresponding record j in B. This is, $\pi(i) = j$ can be understood as the linkage of individual i in file A with the individual j in file B.

Using this notation, we can compute the following:

1. The number of possible re-identifications: $n!$
2. The number of permutations such that there are exactly r elements correctly re-identified: These permutations with exactly r correct links can be built considering the following steps (we use below $k := n - r$). First, we take k elements from the correct permutation π and we permutate this k elements in such a way that there is no one that keeps its original position. Therefore, there are exactly r elements correctly re-identified (the elements not selected). To compute the cardinality of this set of permutations, we need to know that the number of sets that can be taken with k elements is: $n!/(k!(n-k)!)$. Also, that a *permutation without fixed point* (see e.g. [25]) generated from π' is a permutation π'' in such a way that there is no element that keeps its original position (i.e., $\pi'(i) \neq \pi''$ for all i). The number of permutations of k elements without fixed

point is [25]:

$$(\hat{p})(k) = k! \sum_{v=0}^{k} \frac{(-1)^k}{v!}$$

According to this, the number of permutations such that there are exactly r elements in the correct position is:

$$\frac{n! \sum_{v=0}^{k} \frac{(-1)^k}{v!}}{(n-k)!}$$

3. The probability of finding at random a permutation with exactly r elements in the correct position:

$$\frac{\sum_{v=0}^{k} \frac{(-1)^k}{v!}}{(n-k)!}$$

In Table V probabilities for the case of having $n = 33$ (the ionosphere case) are given. Note that the probability of obtaining 10 or more correct links (the ones obtained in the example above for the ionosphere data file) is $1.01377715E-7$. Similarly, Table VI displays the probabilities for the case of $n = 8$. This is the case of the abalone data file. In this case, the probability of having more than 6 correct links (the ones obtained in this example) is $7.1924605E-4$. Table VII displays the corresponding probabilities for the iris file ($n = 4$).

An additional aspect to be considered for the evaluation of this result is that standard techniques for record-linkage when files share a set of common variables do not lead to 100% re-identifications. In fact, [41] describe about 300 record-linkage experiments (one third using numerical data and the rest using categorical data) and the corresponding percentage of re-identifications for both probabilistic and distance-based record linkage approaches are listed. The following averages can be computed from the listings: 26.12% (for distance-based record linkage for numerical data), 19.72% (for probabilistic one for numerical data), 59.30% (for distance-based one for categorical data) and 57.93% (for probabilistic one for categorical data). In the approach presented here, we got 75% re-identifications for the abalone file and 30.30% for the ionosphere file.

According to all this, the abalone and ionosphere examples show that although re-identification for files not sharing variables is far from perfect, the approach proposed here is appropriate because it obtains meaningful and relevant links.

VI. Conclusions.

In this work we have considered the use of aggregation operators in data mining. We have considered two different uses. First, we have shown its application in the process of building data models. We have argued that some of the learning techniques are appliable to large databases because the cost is proportional to the number of variables. An example that considers 33 variables has been given. Second, we have shown that aggregation operators are useful for extracting implicit information from raw data and we have applied them to a re-identification problem. We have shown that OWA operators are suitable in this task. We have compared our approach with random selection of linked pairs and with the results usual in re-identification using standard approaches. In both comparisons, our method is well rated and, thus, it shows that the method is suitable for re-identification for files not sharing variables.

r	probability $\|links\| = r$	probability $\|links\| \geq r$
0	0.36787942	1.0
1	0.36787942	0.63212055
2	0.18393971	0.26424113
3	0.06131324	0.0803014
4	0.01532831	0.018988157
5	0.003065662	0.0036598467
6	5.109437E-4	5.941848E-4
7	7.299195E-5	8.3241146E-5
8	9.123994E-6	1.0249197E-5
9	1.0137771E-6	1.1252026E-6
10	1.01377715E-7	1.1142548E-7
11	9.216156E-9	1.0047766E-8
12	7.680129E-10	8.316107E-10
13	5.907792E-11	6.359777E-11
14	4.2198515E-12	4.5198524E-12
15	2.8132344E-13	3.0000107E-13
16	1.7582715E-14	1.8677634E-14
17	1.0342773E-15	1.0949201E-15
18	5.745985E-17	6.064281E-17
19	3.0242027E-18	3.1829554E-18
20	1.5121014E-19	1.5875276E-19
21	7.200482E-21	7.5426254E-21
22	3.2729465E-22	3.4214245E-22
23	1.4230203E-23	1.4847793E-23
24	5.929247E-25	6.1758905E-25
25	2.3717165E-26	2.4664351E-26
26	9.121372E-28	9.471847E-28
27	3.3801082E-29	3.5047438E-29
28	1.202626E-30	1.246356E-30
29	4.241236E-32	4.3729944E-32
30	1.2566625E-33	1.317584E-33
31	6.080625E-35	6.092142E-35
32	0	1.1516336E-37
33	1.1516335E-37	1.1516336E-37

TABLE V

Probabilities of having r correct links, and of having more or equal than r links for 33 individuals.

Acknowledgments

Partial support of the European Community (contract no. IST-2000-25069), of the U.S. Census Bureau (contract no. OBLIG-2000-029144-0-0) and of MCyt (Contract no. TIC2001-0633-C03-02) is acknowledged.

References

[1] Bacher, J., Brand R., Bender, S., (2002), Re-identifying register data by survey data using cluster analysis: an empirical study,

r	probability $\|links\| = r$	probability $\|links\| \geq r$
0	0.36788195	1.0
1	0.36785713	0.63211805
2	0.18402778	0.26426092
3	0.06111111	0.080233134
4	0.015625	0.019122023
5	0.0027777778	0.0034970238
6	6.9444446E-4	7.1924605E-4
7	0	2.4801588E-5
8	2.4801588E-5	2.4801588E-5

TABLE VI

Probabilities of having r correct links, and of having more or equal than r links for 10 individuals.

r	probability $\|links\| = r$	probability $\|links\| \geq r$
0	0.375	1.0
1	0.333333	0.625
2	0.25	0.291666
3	0.0	0.041666
4	0.041666	0.041666

TABLE VII

Probabilities of having r correct links, and of having more or equal than r links for 4 individuals.

International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10:5, 589-608.

[2] Bauer, E., Kohavi, R., (1999), An Empiricial Comparison of Voting Classification Algorithms: Bagging, Boosting and Variants, Machine Learning 36 105-139.

[3] Beliakov, G., (2003), How to build aggregation operators from data, Int. J. of Intel. Systems, 18 903-923.

[4] Calvo, T., Mayor, G., Mesiar, R., (2002), Eds., Aggregation operators: New trends and applications, Physica-Verlag, Springer.

[5] Carbonell, M., Mas, M., Mayor, G., (1997), On a class of Monotonic Extended OWA Operators, Proc. of the Sixth IEEE International Conference on Fuzzy Systems (IEEE-FUZZ'97), 1695-1699, Barcelona, Catalonia, Spain.

[6] Dempster, A. P., Laird, N. M., Rubin, D. B., (1977), Maximum Likelihood From Incomplete Data Via the EM Algorithm, *Journal of the Royal Statistical Society*, 39, 1-38.

[7] Domingo-Ferrer, J., Torra, V., (2003), On the Connections Between Statistical Disclosure Control for Microdata and Some Artificial Intelligence Tools, Information Sciences, in press.

[8] Domingo-Ferrer, J., Torra, V., (2003), Disclosure Risk Assessment in Statistical Microdata Protection via Advanced Record Linkage, Statistics and Computing, 13 (2003) 343-354.

[9] Filev, D. P., Yager, R. R., (1994), Learning OWA operator weights from data, Proc. of the 3rd IEEE Conf. on Fuzzy Systems, Orlando, 468-473.

[10] Filev, D. P., Yager, R. R., (1998), On the issue of obtaining OWA operator weights, Fuzzy Sets and Systems, 94 157–169.

[11] Gill, L., (2001), *Methods for Automatic Record Matching and Linking and Their Use in National Statistics*, National Statistics Methodology Series no. 25, London: Office for National Statistics.

[12] Grabisch, M., Nguyen, H. T., Walker, E.A., (1995), Fundamentals of Uncertainty Calculi with Applications to Fuzzy Inference, Kluwer Academic Publishers, Dordreht, The Netherlands.

[13] Grabisch, M., (2003), Modelling data by the Choquet integral, V. Torra (Ed.), Information Fusion in Data Mining, Springer, 135-148.

[14] http://www.integrity.com

[15] Jaro, M. A., (1989), Advances in record-linkage methodology as applied to matching the 1985 Census of Tampa, Florida, *Journal of the American Statistical Association*, 84:406, 414-420.

[16] Merz, C. J., (1999), Using Correspondence Analysis to Combine Classifiers, Machine Learning, 36 33-58.

[17] Merz, C. J., Pazzani, M. J., (1999), Combining regression estimates, Machine learning, 36 9-32.

[18] Murphy, P., M., Aha, D. W., (1994), UCI Repository machine learning databases, http://www.ics.uci.edu/ mlearn/MLRepository.html, Irvine, CA: University of California, Department of Information and Computer Science.

[19] Sugeno, M., Murofushi, T., (1993), Fuzzy Measures (in Japanese), Nikkankogyou, Tokyo, Japan.

[20] Nettleton, D., Muniz, J., (2001), Processing and representation of meta-data for sleep apnea diagnosis with an artificial intelligence approach, Int. J. of Medical Informatics, 63, 77 - 89

[21] Newcombe, H. B., Kennedy, J.M., Axford, S. J., James, A.P, (1959), Automatic Linkage of Vital Records, Science, 130, 954-959.

[22] O'Hagan, M., (1988), Aggregating template or rule antecedents in real-time expert systems with fuzzy set logic, Proc. of the 22nd Annual IEEE Asilomar Conf. on Signals, Systems and Computers, Pacific Grove, CA, 681–689.

[23] Pagliuca, D., Seri, G., (1999), *Some Results of Individual Ranking Method on the System of Enterprise Accounts Annual Survey*, Esprit SDC Project, Deliverable MI-3/D2.

[24] Rahm, E., Do, H. H., (2000), Data cleaning: problems and current approaches, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 23:4 3-13.

[25] Reinhard, F., Soeder, H., (1997), Atlas des mathématiques, Librairie Générale Française.

[26] Robinson-Cox, J. F. (1998), A record-linkage approach to imputation of missing data: analyzing tag retention in a tag-recapture experiment, J. of Agricultural, Bilogical, and Environmental Statistics, Vol. 3: 1, pp. 48-61.

[27] Ribeiro, R. A., (1996), Fuzzy multiple attribute decision making: A review and new preference elicitation techniques, Fuzzy Sets and Systems, 78:2 155–181.

[28] Saaty, T. L., (1980), The Analytic Hierarchy Process, McGraw-Hill, New York.

[29] Sigillito, V. G., Wing, S. P., Hutton, L. V., Baker, K. B. (1989), Classification of radar returns from the ionosphere using neural networks. John Hopkins APL Technical Digest, 10, 262–266.

[30] Sugeno, M., (1974), Theory of Fuzzy Integrals and its Applications. (PhD Dissertation). Tokyo Institute of Technology, Tokyo, Japan.

[31] Tanaka, A., Murofushi, T., (1989), A learning model using fuzzy measures and the Choquet integral, 5th Fuzzy System Symposium, Kobe, Japan, 213–218 (in japanese).

[32] Torra, V., (1997), The Weighted OWA operator, Int. J. of Intel. Syst., 12:2 153–166.

[33] Torra, V., (1999), On the learning of weights in some aggregation operators: the weighted mean and OWA operators, Mathware and Soft Computing, 6, 249–265.

[34] Torra, V., (1999), On Some Relationships between Hierarchies of Quasiarithmetic Means and Neural Networks, Int. J. of Intel. Systems 14:11 1089–1098.

[35] Torra, V., (2000), Towards the re-identification of individuals in data files with non-common variables, Proceedings of the European Conference on Artificial Intelligence (ECAI 2000), 326-330, Berlin, Germany.

[36] Torra, V., (2000), The WOWA operator and the interpolation function W*: Chen and Otto's interpolation method revisited, Fuzzy Sets and Systems, 113:3 389-396.

[37] Torra, V., (2000), Re-identifying Individuals using OWA operators, Proc. of the 6th Int. Conference on Soft Computing (Iizuka 2000) (ISBN 4-938717-05-0), 101 (full paper in the electronic proceedings - CD Rom), Iizuka, Fukuoka, Japan, 2000.

[38] Torra, V., (2000), Learning Weights for Weighted OWA Operators, Proc. of the IEEE Int. Conference on Industrial Electronics, Control and Instrumentation (IECON 2000), Nagoya, Japan. (CD-ROM).

[39] Torra, V., (2002), Learning weights for the quasi-weighted mean, IEEE Trans. on Fuzzy Systems, 10:5 653-666.

[40] Torra, V., (2003), Information fusion in data mining, Springer.

[41] Torra, V., Domingo-Ferrer, J., (2003), Record linkage methods for multidatabase data mining, V. Torra (Ed.), Information Fusion in Data Mining, Springer, 101-132.

[42] Tumer, K., Ghosh, J., (1996), Classifier Combining: Analytical Results and Implications. Working notes from the Workshop Integrating Multiple Learned Models, National Conference on Artificial Intelligence, August 1996, Portland. `http://ic.arc.nasa.gov/~kagan/tumer_pubs.html`

[43] U. S. Bureau of the Census (2000), Record Linkage Software: User Documentation, Available from U. S. Bureau of the Census.

[44] Winkler, W. E., (1995), Advanced methods for record linkage, American Statistical Association, Proceedings of the Section on Survey Research Methods, pp. 467-472.

[45] Winkler, W.E., (1995), Matching and Record Linkage, in B.G. Cox (ed.), Business Survey Methods, New York, J. Wiley, 355-384.

[46] Xu, Z. S., Da, Q. L., (2003), An overview of operators for aggregating information, Int. J. of Intel. Systems 18 953-969.

[47] Yager, R. R., (1988), On ordered weighted averaging aggregation operators in multi-criteria decision making, IEEE Trans. on SMC, 18 183-190.

[48] Yager, R.R., (1993), Families of OWA operators, Fuzzy Sets and Systems, 59 125-148.

[49] Yager, R. R., (1996), Quantifier Guided Aggregation Using OWA operators, Int. J. of Int. Systems, 11 (1996) 49-73.

[50] Zhong, N., Yao, Y., Ohsuga, S., (1999), Peculiarity oriented multidatabase mining, J. Zytkow, J. Rauch, (Eds.), Principles of Data Mining and Knowledge Discovery, Lecture Notes in Artificial Intelligence 1704, Springer, 136-146.