

Deakin Research Online

This is the published version:

Beliakov, Gleb and Calvo, Tomasa 2005, Identification of general and double aggregation operators using monotone smoothing, in *EUSFLAT 2005 : Proceedings of the Fourth Conference of the European Society for Fuzzy Logic and Technology and 11 Rencontres Francophones sur la Logique Floue et ses Applications*, EUSFLAT, Barcelona, Spain, pp. 937-942.

Available from Deakin Research Online:

<http://hdl.handle.net/10536/DRO/DU:30005743>

Every reasonable effort has been made to ensure that permission has been obtained for items included in Deakin Research Online. If you believe that your rights have been infringed by this repository, please contact drosupport@deakin.edu.au

Copyright : 2005, EUSFLAT

Identification of general and double aggregation operators using monotone smoothing

Gleb Beliakov

School of Information Technology,
Deakin University, 221 Burwood Hwy,
Burwood 3125, Australia
gleb@deakin.edu.au

Tomas Calvo

Departamento de Ciencias de la Computación
Campus Universidad de Alcalá
28871-Alcalá de Henares (Madrid), Spain
tomasa.calvo@uah.es

Abstract

Aggregation operators model various operations on fuzzy sets, such as conjunction, disjunction and averaging. Recently double aggregation operators have been introduced; they model multistep aggregation process. The choice of aggregation operators depends on the particular problem, and can be done by fitting the operator to empirical data. We examine fitting general aggregation operators by using a new method of monotone Lipschitz smoothing. We study various boundary conditions and constraints which determine specific types of aggregation.

Keywords: Aggregation operators, Empirical fit, Monotone approximation.

1 Introduction

Aggregation operators model various fuzzy logic operations, such as conjunction, disjunction, averaging, as well as their combinations. Many families of aggregation operators, such as triangular norms and conorms, uninorms, ordered weighted averaging (OWA), generalized means and many others have been extensively studied in fuzzy sets literature, see [1, 2]. Frequently this extensive range is not sufficient for applications, and new types of aggregation operators are developed. One recent example is that of double aggregation operators, that model two-step aggregation procedures [3]. In this process some membership values (arguments) are combined using one operator, other arguments are combined using a different operator, and at the second stage the outcomes are combined with a third operator. These opera-

tors can model the following logical constructions

If $(A \text{ AND } B \text{ AND } C) \text{ OR } (D \text{ AND } E)$ then

for example, by using two triangular norms for **AND** and a triangular conorm for **OR**. There are also possible generalizations for multistep aggregation processes (see Remark 1 in [3]), but these are beyond the scope of this paper.

On the other hand, given such a variety of aggregation operators, the choice of a particular operator for a particular application is complicated. The method of empirical fit was introduced by Zimmermann and Zysno [4]; it involves fitting the parameters of an aggregation operator to empirical data, as to approximate these data best. More recently, Filev and Yager [5] considered the problem of fitting OWA operators to the data, Dyckhoff and Pedrycz [6] considered this problem for generalized means, and Beliakov presented methods applicable to general aggregation operators, associative operators, OWA and generalized means [7, 8, 9].

One should be aware that aggregation operators are special functions, and they require specially tailored regression techniques to be fitted to the data. In this paper we examine fitting general aggregation operators, which is the most broad and flexible class of aggregation operators. We shall use the method of optimal Lipschitz interpolation and smoothing, recently applied to aggregation operators in [10]. We shall concentrate on various types of inequality constraints one can impose on the aggregation operator, and on how these constraints translate into the properties of operators.

2 Problem formulation

We define a general aggregation operator as an n -variate monotone increasing function $f : [0, 1]^n \rightarrow [0, 1]$, satisfying $f(\mathbf{0}) = 0$, $f(\mathbf{1}) = 1$. Note that in [2], the term "general" aggregation operator refers to a family of n -variate operators, $n = 2, 3, \dots$. Here we study one operator from such a family of a fixed dimension n . Methods of identification of the whole family are presented in [9, 11].

Monotonicity of aggregation operators is semantically important, and must be preserved during the fitting process. Besides monotonicity, there could be other restrictions, such as commutativity, idempotency, or disjunctive/conjunctive behaviour. Moreover these restrictions sometimes apply only on parts of the domain (e.g., uninorms and nullnorms change disjunctive behaviour to conjunctive within their domain). These restrictions usually come from the domain-specific knowledge and specific problem requirements. Our goal is to translate these restrictions to the constraints that could be incorporated into the fitting process, so that to obtain operators with exactly the required properties.

Consider the problem of fitting a general aggregation operator f to empirical data in the form

$$D = \{(x_1^k, x_2^k, \dots, x_n^k), y^k\}, k = 1, 2, \dots, K. \quad (1)$$

There are K observations, and the k -th observation has n observed arguments and the observed aggregated value y^k . Such observations may come from an experiment, such as in [4], or be the desired output of a fuzzy system in response to given vector of arguments. We are after a monotone increasing function f , such that

$$f(x_1^k, x_2^k, \dots, x_n^k) \approx y^k, k = 1, 2, \dots, K. \quad (2)$$

For general aggregation operators, one approach is to use monotone tensor product splines, which is described in [7]. However, because of the curse of dimensionality, tensor product schemata are applicable only to small n , as the number of basis functions (and coefficients to compute) grows as an exponent of n .

The empirical data has the following properties: a) the data are scattered; b) the observed values may contain errors; c) the data may not be

monotone. The two latter properties require a smoothing process, as to make it compatible with the desired class of functions. We later show how this can be done by using linear or quadratic programming techniques.

3 Lipschitz approximation

Suppose that we have the data set D (possibly smoothed) which we want to interpolate, and the requirement that the interpolating function be monotone and Lipschitz continuous, with Lipschitz constant M . We remind that Lipschitz continuity is expressed as

$$\exists M : \forall x, z, |f(x) - f(z)| \leq M\|x - z\|,$$

the smallest such constant is called the Lipschitz constant of f and is denoted by $L(f)$. We are interested in the following classes of functions

$$Lip(M) = \{f : L(f) \leq M\} \text{ and}$$

$$Mon = \{f : x \preceq z \Rightarrow f(x) \leq f(z)\},$$

where $x \preceq z$ means $\forall i \in \{1, \dots, n\} : x_i \leq z_i$.

The condition $f \in Lip(M) \cap Mon$ restricts the values of f to the following bounds

$$H^{lower}(x) \leq f(x) \leq H^{upper}(x),$$

$$H^{upper}(x) = \min_k \{y^k + M\|(x - x^k)_+\| \},$$

$$H^{lower}(x) = \max_k \{y^k - M\|(x^k - x)_+\| \}, \quad (3)$$

where $(t)_+ = \max\{t, 0\}$, which is applied componentwise for vectors. The optimal interpolant is the one which minimizes the error of approximation in the worst case scenario; it is given as

$$g(x) = \frac{1}{2}(H^{upper}(x) + H^{lower}(x)). \quad (4)$$

Note that $H^{upper}, H^{lower}, g \in Lip(M) \cap Mon$.

The existence of the optimal interpolant, as well as the existence of any interpolant from $Lip(M) \cap Mon$, depends on whether the data is consistent with both monotonicity and the Lipschitz condition. One can prove that it is consistent if and only if the following inequalities hold

$$\forall i, j \in \{1, \dots, K\} : y^i - y^j \leq M\|(x^i - x^j)_+\|. \quad (5)$$

Because of inaccuracies in the data, it may be inconsistent with the class $Lip(M) \cap Mon$, and in this case it needs to be smoothened. Let \tilde{y}^k denote the smoothened data, compatible with the desired class, and $r^k = \tilde{y}^k - y^k$ denote the residuals. Then we can smooth the data by minimizing the norm of the residuals, by solving

$$\begin{aligned} & \min \sum_{k=1}^K |r^k|^p, \\ \text{s.t. } & r^i - r^j \leq y^j - y^i + M\|(x^i - x^j)_+\|. \end{aligned} \quad (6)$$

If we choose $p = 2$, we minimize the least squares criterion, subject to the linear constraints, which is a standard quadratic programming problem. For $p = 1$ we obtain the least absolute deviation problem, frequently used in robust regression, as it is less sensitive to outliers. In this case by splitting r^k into positive and negative parts, and using $r^k = r_+^k - r_-^k$, $|r^k| = r_+^k + r_-^k$, $r_+^k, r_-^k \geq 0$ we transform (6) into a linear programming problem.

Thus the process of monotone Lipschitz smoothening consists of two steps. First one solves problem (6) for r^k by using quadratic or linear programming techniques. Then, once the data is smoothened, the function $g \in Lip(M) \cap Mon$ which approximates the data best is computed from (3),(4), where y^k are substituted with \tilde{y}^k .

4 Fitting general aggregation operators

To fit a general aggregation operator to noiseless data one proceeds as follows. Firstly one needs to identify the Lipschitz constant M of the operator, i.e., to specify the class $Lip(M)$. This can be done using background information about the problem and specific practical requirements. If no such information is available, one can determine the smallest class $Lip(M)$ still consistent with the data set D , by choosing

$$M = \inf\{C > 0 : y^i - y^j \leq C\|(x^i - x^j)_+\|\}.$$

This is done by direct computation.

To guarantee $f(\mathbf{0}) = 0, f(\mathbf{1}) = 1$, we add these equations in the form of interpolation conditions, i.e., by augmenting the data set D with these two data. Then Eqs. (3),(4) yield the optimal aggregation operator from $Lip(M) \cap Mon$, which interpolates the data best in the worst case scenario.

For noisy data, we also have to identify the class $Lip(M)$. This can be done by using problem specific information, or calculated automatically from the data. The latter method requires either data splitting or cross-validation techniques, which we will not discuss here. Then we add $f(\mathbf{0}) = 0, f(\mathbf{1}) = 1$ as the additional interpolation conditions. We solve problem (6) to obtain the smoothened data \tilde{y}^k (the additional interpolation conditions are added to the set of linear constraints). Then Eqs. (3),(4) yield the optimal interpolant, with y^k substituted with \tilde{y}^k .

These approximation processes are also followed in the case of special classes of operators, but with a number of additional constraints, as described below. The new constraints can be incorporated in several ways: as linear constraints in (6), as modifications of (3), or implicitly as extra interpolation conditions. In each case we aim at choosing the most simple method.

5 Special classes of operators

In this section we consider four types of aggregation operators: disjunctive, conjunctive and averaging operators, as well as commutative operators. A conjunctive aggregation operator models fuzzy *AND* and is associated with the restriction $f(t, 1) \leq t$. The typical representatives are triangular norms. A disjunctive operator models fuzzy *OR*, and is associated with $f(t, 0) \geq t$. The typical representatives are triangular conorms. When the equalities hold, we talk about the neutral element $e \in [0, 1] : \forall t \in [0, 1], f(t, e) = t$, which is 1 in the first case and 0 in the second. The neutral element e also plays a similar role in bi-polar aggregation operators.

In vector form (for $n > 2$; we do not assume associativity here), the above conditions are written as $\forall i \in \{1, \dots, n\} : f(\mathbf{e}(t, i)) = t$, where $\mathbf{e}(t, i) = (e, \dots, e, t, e, \dots, e)$ and t is in the i -th position. Averaging operators are associated with the idempotency $f(t, t, \dots, t) = t$.

It is not difficult to see that the above properties, together with the monotonicity translate into the following restrictions on the whole domain.

- Conjunctive behavior implies $f \leq \min$.

- Disjunctive behavior implies $f \geq \max$.
- Idempotency implies $\min \leq f \leq \max$.

Thus if the aggregation operator is supposed to have conjunctive behaviour, we change both bounds (3) into

$$\begin{aligned} H_{conj}^{upper}(x) &= \min\{H^{upper}(x), \min(x)\} \\ H_{conj}^{lower}(x) &= \min\{H^{lower}(x), \min(x)\}, \end{aligned} \quad (7)$$

where $\min(x) = \min\{x_1, x_2, \dots, x_n\}$. The other cases are dealt with similarly, and (4) applies.

If one needs to obtain the actual equality $f(\mathbf{e}(t, i)) = t$, then the following extra bounds are required

$$\begin{aligned} \forall i \in \{1, \dots, n\}, \forall t \in [0, 1], \forall z \in [0, 1]^n, z_i = t : \\ B_i^l(z) \leq f(z) \leq B_i^u(z), \quad (8) \\ B_i^l(z) = t - M\|\mathbf{e}(t, i) - z\|, \\ B_i^u(z) = t + M\|\mathbf{e}(t, i) - z\|. \end{aligned}$$

These bounds directly follow from the Lipschitz condition, and are added to (9).

Consider now the case of an aggregation operator whose behaviour changes inside its domain. Uninorms and nullnorms are typical representatives of this class of operators. Consider first the case of the neutral element $0 < e < 1$, as in uninorms. Then we have conjunctive behaviour on $[0, e]^n$, disjunctive behaviour on $[e, 1]^n$, and unrestricted on the rest of the domain.

We immediately have $\forall x \in [0, e]^n : f(x) \leq \min(x)$, $\forall x \in [e, 1]^n : f(x) \geq \max(x)$. Further, we also need conditions (8). We incorporate them as

$$\begin{aligned} H_{mix}^{upper}(x) &= \min\{H^{upper}(x), B^u(x)\} \\ H_{mix}^{lower}(x) &= \max\{H^{lower}(x), B^l(x)\}, \end{aligned} \quad (9)$$

where

$$B^u(x) = \min\{\min_i\{B_i^u(x)\}, \min_{[0,e]}(x)\},$$

$$B^l(x) = \max\{\max_i\{B_i^l(x)\}, \max_{[e,1]}(x)\},$$

$$\min_{[0,e]}(x) = \min_i(x_i) \text{ if } x \in [0, e]^n, 1 \text{ otherwise,}$$

$$\max_{[e,1]}(x) = \max_i(x_i) \text{ if } x \in [e, 1]^n, 0 \text{ otherwise.}$$

An aggregation operator is said to have an annihilator $a \in [0, 1]$, if $\forall x : f(x, a) = a$. For more than two arguments the formula extends as $f(\mathbf{a}(x, i)) = a$, where $\mathbf{a}(x, i) = (x_1, \dots, x_{i-1}, a, x_{i+1}, \dots, x_n)$. The existence of an annihilator does not imply conjunctive or disjunctive behaviour on any part of the domain, but together with monotonicity, it implies $f(x) = a$ on $[a, 1] \times [0, a]$ and $[0, a] \times [a, 1]$ (and their multivariate extensions).

These restrictions are easily incorporated into the bounds by using

$$\begin{aligned} \max_i B_i^l(x) \leq f(x) \leq \min_i B_i^u(x), \quad (10) \\ B_i^l(x) = a - M\|\mathbf{a}(x, i) - x\|_+, \\ B_i^u(x) = a + M\|(x - \mathbf{a}(x, i))_+\|. \end{aligned}$$

For aggregation operators that are disjunctive on $[0, a]^n$ and conjunctive on $[a, 1]^n$ (a typical example is nullnorms, although we do not require associativity), we add the constraints

$$\max_{[0,a]}(x) \leq f(x) \leq \min_{[a,1]}(x),$$

where the restricted minimum and maximum are calculated as in (9).

Of course, there are many variations of the above restrictions, which can apply on different parts of the domain. These restrictions lead to similar modifications to the upper and lower bounds in (3), making these bounds tighter. An important point is that these restrictions involve max/min and linear functions, and are easily computable. This means that in the case of noisy data, when we need to solve the smoothing problem like (6), the new restrictions will simply augment the set of constraints in (6), but will not modify the type of the optimization problem. Thus we will have a quadratic or linear programming problem, with a larger set of inequality constraints.

To finish this section, consider a different type of a priori information, that the aggregation operator is commutative $f(x) = f(x_P)$, where P is any permutation of indices. This requirement may be in addition to the previous types of restrictions. The way to enforce commutativity is to approximate the function $f_{\emptyset}(x)$, defined on the simplex $S = \{x : x_1 \leq x_2 \leq \dots \leq x_n\}$. Then

the commutative aggregation operator is found as $f(x) = f_{\langle} (x_{\langle})$, where x_{\langle} denotes the vector obtained from x by sorting its components in increasing order. To approximate f_{\langle} we use the same method of Lipschitz approximation, but apply it to the data set $D_{\langle} = \{x_{\langle}^k, y^k\}$. The other restrictions specifying conjunctive, disjunctive or averaging behaviour need no modification.

6 Double aggregation operators

Double aggregation operators were introduced in [3] with the purpose to model multistage aggregation process. They are defined as

$$f(x) = F(G(p), H(q)),$$

where F, G, H are aggregation operators, and $p \in [0, 1]^k, q \in [0, 1]^m, k + m = n$ and $x = p|q$. " $\cdot|$ " denotes concatenation of two vectors. A typical application of such operators is when the information contained in p and q is of different nature, and is aggregated in different ways. F may have more than two arguments.

While the resulting operator f is a general aggregation operator, and as such can be fitted to the data as described in section 4, we may have more specific information about the operators F, G, H , and we shall now use such information. The main challenge here is that the operator f will not generally share the properties of F, G, H [3]. For instance, the fact that G is conjunctive does not imply conjunctive nature of f . The same applies to other properties.

Consider some special cases, in which information about the properties of F, G, H translates into suitable restrictions on f . The simplest case is when some of these operators are fixed a priori. For instance suppose that operators G and H are fixed. Then we can use the method from section 4 to approximate f , with $G(p)$ and $H(q)$ playing the role of x^1, x^2 .

Let us now consider the case when all operators F, G, H need to be approximated. Here we can use the following properties:

- If the aggregation operators F, G, H are all conjunctive, then f is conjunctive.

- If the aggregation operators F, G, H are all disjunctive, then f is disjunctive.
- If the aggregation operators F, G, H are all idempotent, then f is idempotent.

The proof is simple. In the first case we have $G(p) \leq \min(p), H(q) \leq \min(q), F(G, H) \leq \min(G, H)$, therefore

$$\begin{aligned} f(x) &\leq \min(G(p), H(q)) \\ &\leq \min(\min(p), \min(q)) = \min(x). \end{aligned}$$

The second case is analogous. In the last case

$$f(t, t, \dots) = F(G(t, \dots), H(t, \dots)) = F(t, t) = t.$$

Then we can apply the methods from section 5.

A more interesting case is when operators F, G, H have different behaviour. We can distinguish the following cases.

- G, H are conjunctive, F is idempotent;
- G, H are disjunctive, F is idempotent;
- G, H are idempotent, F is conjunctive;
- G, H are idempotent, F is disjunctive;
- G is idempotent, F, H are conjunctive;
- G is idempotent, F, H are disjunctive.

In the case (a) we have

$$f(x) \leq \max(G, H) \leq \max(\min(p), \min(q)),$$

and in the case (b)

$$\min(\max(p), \max(q)) \leq \min(G, H) \leq f(x).$$

Case (c) translates into

$$f(x) \leq \min(G, H) \leq \min(\max(p), \max(q)),$$

and case (d) translates into

$$\max(\min(p), \min(q)) \leq \max(G, H) \leq f(x).$$

For (e),(f) we have

$$f(x) \leq \min(G, H) \leq \min(\max(p), \min(q)),$$

$$\max(\min(p), \max(q)) \leq \max(G, H) \leq f(x).$$

Lastly, consider commutativity of F, G, H . Double aggregation operator will be commutative if and only if F, G, H are commutative and $G = H$ [3]. Such conditions may be too strong for many applications. If they hold partially, then we call the aggregation operator symmetric (right- or left-symmetric). For example, if G is commutative, then f is left-symmetric.

To incorporate the symmetry, we consider permutations of vectors $p_{()}, q_{()}$, in which the elements are arranged in increasing order. We approximate the function $f_{(,), (,)}(x) = f(p_{()}, q_{()})$, defined on the Cartesian product of two simplices $S_p = \{p \in [0, 1]^k : p_1 \leq \dots \leq p_k\}$, $S_q = \{q \in [0, 1]^m : q_1 \leq \dots \leq q_m\}$. We proceed as at the end of section 5, by constructing the data set $D_{(,), (,)} = \{(p_{()}^i, q_{()}^i, y^i), i = 1, \dots, K\}$, and approximating $f_{(,), (,)}(x)$. If we only need left/right symmetry, we use a permutation of either p or q .

7 Conclusion

We considered optimal approximation of general aggregation operators by monotone Lipschitz functions. We identified various types of restrictions on aggregation operators, and translated these restrictions into tight upper and lower bounds on f . When one global value of the Lipschitz constant M is not adequate, approximation of locally Lipschitz functions can be used. In this case different parts of the domain will have different Lipschitz constants (which can be identified from the data). Our work on locally Lipschitz functions will be reported elsewhere.

Acknowledgements

This work was supported by the projects MTM2004-3175 from Ministerio de Educacion y Ciencia, Spain, Cost 274 Tarski, PRIB-2004-9250, Govern de les Illes Balears, and BFM2003-05308.

References

[1] D. Dubois and H. Prade. A review of fuzzy set aggregation connectives. *Information Sciences*, 36:85–121, 1985.

- [2] T. Calvo, A. Kolesarova, M. Komornikova, and R. Mesiar. Aggregation operators: properties, classes and construction methods. In T. Calvo, G. Mayor, R. Mesiar, eds, *Aggregation Operators. New Trends and Applications*, pp. 3–104. Physica-Verlag, Heidelberg, New York, 2002.
- [3] T. Calvo and A. Pradera. Double aggregation operators. *Fuzzy Sets and Systems*, 142:15–33, 2004.
- [4] H.-J. Zimmermann and P. Zysno. Latent connectives in human decision making. *Fuzzy Sets and Systems*, 4:37–51, 1980.
- [5] D. Filev and R. Yager. On the issue of obtaining OWA operator weights. *Fuzzy Sets and Systems*, 94:157–169, 1998.
- [6] H. Dyckhoff and W. Pedrycz. Generalized means as model of compensative connectives. *Fuzzy Sets and Systems*, 14:143–154, 1984.
- [7] G. Beliakov. How to build aggregation operators from data? *Int. J. Intelligent Systems*, 18:903–923, 2003.
- [8] G. Beliakov. Monotone approximation of aggregation operators using least squares splines. *Int. J. Uncertainty, Fuzziness and Knowledge-Based Systems*, 10:659–676, 2002.
- [9] G. Beliakov, R. Mesiar, and L. Valaskova. Fitting generated aggregation operators to empirical data. *Int. J. Uncertainty, Fuzziness and Knowledge-Based Systems*, 12:219–236, 2004.
- [10] G. Beliakov. Identification of general aggregation operators by Lipschitz approximation. In M.H. Hamza, editor, *The IASTED Intl. Conf. on AI and Appl.*, pp. 230–233, Innsbruck, Austria, 2005. ACTA Press.
- [11] G. Beliakov. Fitting triangular norms to empirical data. In P.E. Klement, R. Mesiar, eds, *Logical, Algebraic, Analytic, and Probabilistic Aspects of Triangular Norms*, pp. 255–265. Elsevier, New York, 2005.