SOBRE FUNCIONES Y REGLAS DE AGREGACIÓN

Javier Montero[1], **Daniel Gómez**[2], **Victoria López**[3], **J. Tinguaro Rodríguez**[1], **Begoña Vitoriano**[1]

[1] Facultad de Matemáticas, Universidad Complutense de Madrid {monty,jtrodrig,bvitoriano}@mat.ucm.es
[2] Escuela de Estadística, Universidad Complutense de Madrid, dagomez@estad.ucm.es
[3] Facultad de Informática, Universidad Complutense de Madrid, vlopez@fdi.ucm.es

Resumen

En este adel lartículo se analiza la definición actualmente al uso de función de agregación, haciendo hincapié en que, a pesar de considerar el problema de la variabilidad de la dimensión de la información a agregar, dicha definición no cierra la posibilidad de disfuncionalidades obvias, subyaciendo en su concepción un potencial abuso del formalismo matemático que, junto con las necesarias consideraciones de implementación práctica pueden llevar a reducir a una simple fórmula lo que es un complejo concepto. En este trabajo propondremos las bases para una definición alternativa que tenga en cuenta los objetivos y las limitaciones que se observan en las aplicaciones de las funciones de agregación dentro del contexto difuso.

Palabras Clave: Funciones de Agregación, Reglas de Agregación, Operadores de Agregación, Recursividad.

1 FUNCIONES DE AGREGACIÓN

La agregación de información es, en sentido amplio, una de las piezas clave de todo sistema de ayuda a la decisión, tanto en cuanto elabora o prepara una visión global de la situación sobre la que se va a tomar una decisión (ver [24]). Con frecuencia esta agregación implica una reducción en la dimensión de la información original, manteniendo sin embargo su formato, aunque esta restricción no es ni mucho menos esencial. De hecho, cabría esperar de manera natural que el grado de complejidad del problema sea superior a la complejidad de cada una de sus descripciones parciales. Así, a veces la mejor descripción de un estado psicológico se logra con una imagen, y una larga estadística numérica se comprende con una gráfica.

En el contexto de operadores difusos, se suele considerar que cada información parcial viene dada en términos de un número dentro del intervalo unidad, para asignar a cada vector de informaciones parciales otro número en el intervalo unidad, que actúa como valor agregado.

De este modo, si nuestro sistema de recogida de datos sabemos alberga indefectiblemente un número determinado de ítems, nuestra agregación puede limitarse a darnos la solución a este problema concreto. Pero en la mayor parte de los casos no podemos garantizar que el cardinal de la información se va a mantener, pues a veces hay datos que se pierden o que deben ser eliminados por errores en la observación o la transmisión, y a veces nos llega información adicional con la que no contábamos, por error en el diseño o por cambios en la realidad observada.

La limitación funcional a una determinada dimensión es sin duda un grave problema. Así, por ejemplo, los operadores OWA son de difícil implementación práctica si mantenemos la definición propuesta por Yager [25], mientras no se aporte un protocolo de actuación ante la pérdida o aparición inesperada de datos [7]. Una propuesta de solución se puede encontrar en [8] (ver también [16]).

Sin embargo, la solución estándar a este problema ha sido con frecuencia mimetizar de manera abusiva la situación en el contexto nítido, donde los valores agregados y la propia agregación toman valores en $\{0,1\}$. Dado que en el caso nítido sólo hay dos operadores razonables (los que corresponden al mínimo y al máximo son los únicos operadores no decrecientes que hacen relevante cada unidad de información recibida), nada más natural que hacerle escoger al decisor entre estos dos contextos (conjunción o disyunción) para aplicar sucesivamente el mismo operador, que es en ambos casos conmutativo y asociativo. Por tanto, podría parecer natural imponer que la estructura básica para analizar una información que toma valores en [0,1] debe estar formada por dos operadores binarios (quizá interconectados con una *negación*) verificando las propiedades de monotonía no decreciente, conmutatividad y asociatividad, además de algunas condiciones de contorno como que la agregación de un vector formado por todo 1 sea 1 o que la agregación de un vector formado por todo 0 sea 0 (ver una discusión de algunas propiedades posibles en [14]). Pero esta aproximación deja fuera de alcance muchas de las agregaciones más habituales, como la media o la mediana. El clásico resultado de Fung y Fu [12], por ejemplo, exige romper con la hipótesis de que toda agregación se puede resolver con un único operador asociativo [22] o hacer una interpretación menos estricta de cuál es el nivel de información en los que los operadores actúan (la media se puede calcular como subproducto de la suma, que sí que resuelve todos sus cálculos a partir de las sumas de parejas de números).

Y es que es obvio que fuera del contexto nítido cada problema de agregación no tiene que poder explicarse mediante la aplicación sucesiva de un mismo operador binario.

Así es como llegamos, dado que tenemos que ser capaces de resolver la agregación sin conocer *a priori* la cardinalidad de los datos, a la definición actualmente estándar de función de agregación: una familia de operadores *A(n)* que sea cuál sea el cardinal *n* de los datos nos diga cómo obtener el valor agregado de esos *n* datos (ver, por ejemplo, [6]).

2 CRITICA A LA DEFINICIÓN ACTUAL

La cuestión que queremos suscitar formalmente en este trabajo es hasta qué punto la definición al uso responde a los fines y propiedades que se esperan de una función de agregación.

Pues si bien es cierto que una condición *sine qua non* para ser función de agregación es que contenga las instrucciones para calcular la agregación de la información, sea cual sea su cardinalidad, hay dos cuestiones a las que dicha definición no responde, y que toda definición debe al menos considerar, toda vez que se ha comprobado su consistencia lógica: su coherencia conceptual y la viabilidad de su implementación práctica.

2.1. COHERENCIA DE UNA FUNCIÓN DE AGREGACIÓN

Por coherencia no nos referimos por supuesto a la consistencia lógica, sin la que no hablaríamos siquiera de definición. Por coherencia entendemos la necesidad de *armonización* entre las distintas componentes de una definición.

Por ejemplo, las ternas de De Morgan (ver por ejemplo [10]) se podrían haber definido como ternas simplemente formadas por una t-norma, una t-conorma y una negación [11], dando así capacidad de resolver en otro contexto las tres operaciones básicas observadas en el caso binario (conjunción, disyunción y negación, y de nuevo hacemos notar que en el caso nítido sólo hay una aplicación que invierta el orden de los dos estados {0,1}, unicidad que no aparece en [0,1], por ejemplo). Pero las ternas de De Morgan se definen imponiendo una propiedad adicional que armoniza esas tres operaciones, como ocurre en el caso nítido (la conjunción de unos datos se puede obtener como la negación de la disyunción de las negaciones de esos datos).

Y es que si no existe una condición adicional que armonice las componentes de una estructura, en realidad no tenemos una estructura. Si una terna de De Morgan estuviese formada simplemente por una t-norma, una t-conorma y una negación, probablemente hablaríamos de una t-norma, una t-conorma y una negación, sin asignar al conjunto un concepto de rango superior.

Sin armonización, sin conexión, sin estructura, no tenemos un edificio, tenemos sólo ladrillos (ver por ejemplo [23] para trasladar a otro contexto la importancia de no obviar el problema de la estructura subyacente).

Del mismo modo, un Espacio de Probabilidad está formado por un espacio muestral, un álgebra de sucesos y una probabilidad, pero estas tres piezas no son arbitrarias sino que tienen que encajar (la probabilidad se define sobre un álgebra de sucesos que tiene que ser isomorfo a un álgebra de subconjuntos del espacio muestral).

Así que, para que una función de agregación merezca ser una definición, debería incorporar alguna propiedad adicional, una restricción que le de un sentido concreto a esa secuencia de operadores *A(2), A(3), A(4),…* , para que se pueda entender que detrás de cada función de agregación hay un concepto unificador.

Parecería cuando menos extraño proponer, por ejemplo, el cálculo del mínimo para dos números, la media aritmética para tres números, la media geométrica para cuatro y la mediana para cinco.

Y tampoco parecería apropiado mantener un enfoque excesivamente formal exigiendo una unidad vía fórmula matemática, pues el ejemplo anterior, por ejemplo, admite una expresión matemática compacta trivial.

2.2. IMPLEMENTACIÓN DE UNA FUNCIÓN DE AGREGACIÓN

El segundo aspecto que toda definición debe tener en cuenta tiene que ver con su implementación. Es decir, la definición debe permitir la existencia de un procedimiento ejecutable que permita obtener el resultado, y en tiempo razonable.

De entrada, es interesante recordar que la función de agregación debe quedar total y unívocamente caracterizada. Y puesto que se trata de una familia infinita de operadores, podría sugerirse de nuevo un enfoque estrictamente formal (determinación de expresiones matemática válidas, para eliminar por ejemplo aquéllas expresiones que implican cálculos conceptualmente diferentes en función del cardinal de los datos).

Pero en cualquier caso acabaremos teniendo que demostrar que los cálculos implicados son asumibles, lo cual sugiere el estudio de sus algoritmos de construcción (desde luego no únicos para una función de agregación, ni necesariamente de la misma complejidad).

Si una definición tiene asociado un algoritmo de cálculo con complejidad exponencial, esa definición no es implementable en la práctica. Una definición de función de agregación no es *operativa* mientras no se haya determinado efectivamente un algoritmo de complejidad polinomial que permita su ejecución real.

Esta característica de *operatividad* fue en su momento el punto de partida para definir lo que en su momento llamamos reglas recursivas [3], que permitían el cálculo a través de la aplicación secuencial de operadores binarios, y donde se entendía por recursividad la capacidad de ejecutar el cálculo ante un dato adicional a partir del resultado anteriormente obtenido, y en las dos direcciones de modo que aparecía subyacente una ecuación general de asociatividad [2,21]. Quizá el resultado más importante con esta aproximación basada en reglas recursivas fue comprobar como bajo determinadas condiciones los primeros operadores podían llegar a determinar toda la secuencia [9]. Pero aunque dicha noción de regla recursiva [3,9] contiene importantes funciones de agregación, es importante hacer notar que el cálculo recursivo puede existir en otros formatos diferentes al propuesto en [3] (la recursividad considerada en las reglas recursivas introducidas en [3] implica un caso específico entre lo que se conoce como algoritmo recursivo), y sobre todo, que en cualquier caso deja fuera otras importantes funciones de agregación. La mediana, por ejemplo, exige con cada nuevo dato reordenar el conjunto de todos los datos, y ninguna reducción de esta información permite el cálculo siguiente (el algoritmo de ordenación añade un factor multiplicativo de complejidad menor que cuadrática y, por lo tanto, sería ejecutable).

3 DEFINICIÓN ALGORÍTMICA DE UNA REGLA DE AGREGACIÓN

De todo lo anterior se justifica el intento de una definición de regla de agregación como una función de agregación basada precisamente en la existencia de un algoritmo apropiado que garantice la comprobación de su coherencia y su cálculo efectivo.

Definición.- Una regla de agregación es una función de agregación que tiene asociada un procedimiento de cálculo específico, que a partir de un número finito de primeros casos, y mediante un algoritmo polinomial iterativo, asigna a todo vector finito de números en el intervalo [0,1] un valor agregado en dicho intervalo.

Nótese que no imponemos condiciones adicionales, ni siquiera la de monotonía (ver por ejemplo [5]). Además, por supuesto que la restricción al intervalo unidad no es ni mucho menos esencial, ni siquiera que todas esas valoraciones sean numéricas, ni que todas ellas tomen valores dentro de un mismo espacio.

En cualquier caso, debe existir al menos un algoritmo de cómputo eficiente y efectivo que resuelva el problema de agregación asociado en tiempo aceptable.

Se trata ésta de una definición de inspiración práctica que intenta distinguir las propuestas con dificultades operativas, con aplicaciones muy específicas y un tanto inusuales. Pudiera parecer una definición casi hueca desde un punto de vista práctico, dado que las funciones de agregación con un algoritmo no polinomial no son operativas, y en general sólo se suele implementar lo que es operativo. Lo que se ha podido implementar es casi por definición polinomial (un algoritmo no polinomial permite, sin embargo, "ocultar" a otros usuarios la recuperación de la información inicial, y por tanto no son siempre desdeñables).

De esta manera, la función de agregación queda caracterizada por un algoritmo de cálculo que se le ha asociado en su definición, y cuyas propiedades se transfieren nominalmente a la función de agregación (iteratividad, recursividad, eficiencia, etcétera). Queda de esta manera habilitada la posibilidad de una clasificación de reglas en función de las propiedades del algoritmo disponible (la búsqueda del algoritmo más eficiente será por supuesto clave). En concreto, nuestras reglas de agregación se han caracterizado en la definición anterior como aquéllas funciones de agregación a las que se les ha podido asociar un algoritmo polinomial.

Así podremos encontrar, por ejemplo, que muchas reglas de agregación necesitan simplemente definir cómo se agregan los dos primeros números. El número de casos necesarios iniciales para “lanzar” el algoritmo puede establecer una primera distinción entre tipos de reglas de agregación. Pero sin duda la característica limitativa está en que a partir de un momento predeterminado se aplique un mismo procedimiento una y otra vez, las veces que sea necesario hasta procesar el conjunto de datos disponible, que será finito en cada caso, aunque sea desconocido *a priori* (quedan muy lejos de nuestro enfoque polinomial aquéllos planteamientos que incluyen la agregación de infinitos datos

Además, la catalogación de una regla de agregación está supeditada al algoritmo disponible (el algoritmo polinomial puede existir y no haberlo encontrado, por ejemplo). Como en complejidad algorítmica (ver [13], por ejemplo), es el análisis del algoritmo concreto, o su asimilación a otro análogo, lo que permite su catalogación como polinomial o no polinomial.

Y tanto la monotonía como la conmutatividad, las condiciones de contorno o las demás propiedades habituales, podrán o no ser convenientes en cada contexto. Como en fiabilidad de sistemas (ver por ejemplo [4]), la mayoría de los sistemas mejoran cuando sus componentes mejoran su comportamiento, pero no siempre (ver de nuevo [14]).

4 COMENTARIOS FINALES

En opinión de los autores, la definición al uso de función de agregación debe ser revisada con la mayor urgencia, dado que considerándola errónea, por estar vacía de contenido real, cuanto más tiempo se tarde más difícil será de desinstalarla. Baste aquí hacer una referencia a lo traumática que ha sido la discusión sobre la utilización de la palabra *intuicionista* en los modelos de Atanassov [1,10] después de contar con miles de referencias (ver también [23]), y que la palabra *difuso* se ha mantenido en castellano para referirse a la lógica *fuzzy*, a pesar de ser lingüísticamente incorrecta, y a pesar incluso de que el propio L.A. Zadeh, fundador de la teoría, ha reconocido en alguna ocasión que cuando se dio cuenta de que la palabra *fuzzy* no había sido una buena elección en lengua inglesa, ya no pudo corregirla porque dicha palabra ya se había instalado totalmente en la comunidad científica.

La propuesta que presentamos está en consonancia con la aproximación *operacional* que se planteaba con la propuesta recursiva de Cutello y Montero [9], generalizando su propuesta y dejando la validez de una regla de agregación en manos de la determinación de un algoritmo polinomial a partir del cual se puedan hacer todos y cada uno de los cálculos que vayan siendo requeridos, y donde se pueden aplicar técnicas de verificación para comprobar su consistencia interna [19,20], garantizando así la presencia de un concepto, a partir de una casuística básica (cómo se agregan los primeros casos) y la repetición sistemática de un procedimiento único a partir de ellos, hasta agotar los datos disponibles o de interés puntual.

Esta aproximación parece contener las funciones de agregación más usuales, abriendo la posibilidad de distinguir casos computacionalmente complejos, introduciendo algunos controles en principio naturales.

El plan de trabajo subsiguiente, toda vez que la clave de nuestra enfoque está en el algoritmo de ejecución de la función de agregación, será trasladar a este concepto todos los matices conocidos en el diseño, análisis y verificación de algoritmos (ver, por ejemplo, [15, 17, 18, 19]).

Pero hemos de resaltar la relevancia de que sea el algoritmo de ejecución quien en última instancia justifica ciertos conceptos asociados a diferentes tipos de función de agregación. Las reglas de agregación en este trabajo definidas, y que destacamos como el campo estándar de trabajo, son nada más que las funciones de agregación polinomiales (es decir, a las que se les ha podido asociar un algoritmo de complejidad temporal polinomial y por lo tanto, eficiente).

Agradecimientos

Este trabajo se ha realizado dentro del proyecto TIN2006-06190 del Plan Nacional de Investigación, y el Grupo de Investigación 910149 de la Universidad Complutense de Madrid.

Referencias

[1] K.T. Atanassov. Intuitionistic fuzzy sets. *Fuzzy Sets and Systems* 20, 87-96, 1986.

[2] J. Aczél. *Lectures on Functional Analysis and their Applications* (Academic Press, New York, 1966).

[3] A. Amo, J. Montero, E. Molina. Representation of consistent recursive rules. *European Journal of Operational Research* 130, 29-53, 2001.

[4] R.E. Barlow, F. Proshan. *Statistical Theory of Reliability and Life Testing* (Holt, Rinehart and Winston, New York, 1975).

[5] G. Beliakov, A. Pradera, T. Calvo. *Aggregation Functions, a Guide to Practitioners* (Springer-Verlag, Berlin, 2007)

[6] T. Calvo, A. Kolesarova, M. Komornikova, R. Mesiar.

Aggregation operators, properties, classes and construction methods. In T. Calvo *et al.* (Eds.): *Aggregation Operators New Trends and Applications* (Physica-Verlag, Heidelberg), 3-104, 2002.

[7] V. Cutello, J. Montero. Hierarchical aggregation of OWA operators: basic measures and related computational problems. *Uncertainty, Fuzzinesss and Knowledge-Based Systems* 3, 17-26, 1995.

[8] V. Cutello, J. Montero. Recursive families of OWA operators. *Proceedings FUZZ-IEEE Conference* (IEEE Press, Piscataway, 1994), 1137-1141.

[9] V. Cutello, J. Montero. Recursive connective rules. *International Journal of Intelligent Systems* 14, 3-20, 1999.

[10] D. Dubois, S. Gottwald, P. Hajek, J. Kacprzyk, H. Prade. Terminological difficulties in fuzzy set theory - the case of intuitionistic fuzzy sets. *Fuzzy Sets and Systems* 156, 485-491, 2005.

[11] J. Fodor, M. Roubens. *Fuzzy Preference Modelling and Mlticriteria Decision Support* (Kluwer, Dordrecht, 1994).

[12] L.W. Fung, K.S. Fu. An axiomatic approach to rational decision making in a fuzzy environment. In: L.A. Zadeh *et at.* (Eds.): *Fuzzy Sets and their Applications to Cognitive and Decision Processes* (Academic Press, New York, 1975), 227-256.

[13] M.R. Garey, D.S. Johnson. *Computer and Intractability, a Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1978).

[14] D. Gómez, J. Montero. A discussion of aggregation functions. *Kybernetika* 40, 107-120, 2004.

[15] C.A.R. Hoare. An axiomatic basis for computer programming. *Communications ACM* 12, 89-100, 1969.

[16] A. Kolesárová. Sequential aggregation. In: M. González *et al.* (Eds.): *Proceedings of the Fifth International Summer School on Aggregation Operators, AGOP* (Universitat de les Illes Balears, Palma de Mallorca, 2009), 183-187.

[17] D. E. Knuth. *The Art of Computer Programming: Fundamental Algorithms*, Vol. 1 (Addison-Wesley, 1968).

[18] D. E. Knuth. Big Omicrom and Big Omega and Big Theta. *SIGACT News, ACM* 8, 18-24, 1976.

[19] V. López, L. Garmendia, J. Montero, G. Resconi. Specification and computing states in fuzzy algorithms. *Uncertainty, Fuzziness and Knowledge-Based Systems* 16, 301-336, 2008.

[20] V. López, J. Montero. Software engineering specification under fuzziness. *Multiple-Valued Logic and Soft Computing* 15, 209-228, 2009.

[21] K.T. Mak. Coherent continuous systems and the generalized functional equation of associativity. *Mathematics of Operations Research* 12, 597-625 (1987).

[22] J. Montero. A note on Fung-Fu`s theorem. *Fuzzy Sets and Systems* 13, 259-269, 1985.

[23] J. Montero, D. Gómez, H. Bustince. On the relevance of some families of fuzzy sets. *Fuzzy Sets and Systems* 158, 2429-2442, 2007.

[24] J. Montero, V. López, D. Gómez. The role of fuzziness in decision making. *Studies in Fuzziness and Soft Computing* 215, 337-349, 2007.

[25] R.R. Yager. On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Transactions on Systems, Man and Cybernetics* 18, 183-190, 1988.