Un estudio sobre el uso de diferentes familias de funciones de fusión para la combinación de clasificadores en la estrategia Uno-contra-Uno

M. Uriz, D. Paternain, H. Bustince, M. Galar
Departamento de Estadística, Informática y Matemáticas, *Universidad Pública de Navarra*,
Campus Arrosadía s/n, 31006 Pamplona, España
{mikelxabier.uriz, daniel.paternain, bustince, mikel.galar}@unavarra.es

***Resumen*—Es este trabajo estudiamos el uso de diferentes familias de funciones fusión para la combinación de clasificadores en un sistema de múltiples clasificadores formado por clasificadores Uno-contra-Uno (del inglés *One-vs-One*, OVO). OVO es una estrategia de descomposición usada para tratar los problemas de clasificación multi-clase, donde el problema original se divide en tantos problemas como pares de clases. En los sistemas de múltiples clasificadores se combinan los clasificadores que provienen de diferentes paradigmas como máquinas de vectores de soporte, algoritmos de inducción de reglas o árboles de decisión. En la literatura, se han desarrollado varios métodos de selección de clasificadores para este tipo de sistemas, donde se busca el clasificador más adecuado para cada par de clases. En este trabajo consideramos el problema desde una perspectiva diferente, con el objetivo de analizar el comportamiento de diferentes familias de funciones fusión para combinar los clasificadores. De hecho, un sistema de múltiples clasificadores OVO puede verse como un problema de toma de decisión multi-experto. En este contexto, para las funciones de fusión que dependen de pesos o medidas difusas, proponemos obtener los parámetros necesarios a partir de los datos. Apoyados en un fuerte análisis experimental, mostramos que la función de fusión utilizada es un factor clave en el sistema final. Además, aquellas funciones basadas en pesos o en medidas difusas pueden permitir modelar mejor el problema de agregación.**

***Index Terms*—Agregaciones, Funciones de fusión, clasificación, One-vs-One, Sistema de múltiples clasificadores**

I. Introducción

En aprendizaje automático, la clasificación consiste en aprender una función (clasificador) utilizando datos etiquetados capaz de asignar la etiqueta correcta a nuevos patrones. Entre los problemas de clasificación se pueden considerar dos escenarios dependiendo del número de clases a distinguir: binario (2 clases) y problemas multi-clase. La clasificación multi-clase generalmente es más difícil ya que la asignación de las fronteras de decisión se vuelve más compleja. Una posible solución para hacer frente a esta dificultad es utilizar estrategias de descomposición [1], donde se divide el problema multi-clase original en problemas binarios más fáciles de resolver. Evidentemente, esta simplificación en la fase de aprendizaje conlleva un coste en la fase de combinación, donde las salidas de todos los clasificadores que se han aprendido en cada nuevo sub-problema deben ser combinados.

Una de las estrategias de descomposición más utilizada es *One-vs-One* (OVO). En OVO, se crean tantos sub-problemas nuevos como pares de clases diferentes, y cada sub-problema es abordado por un clasificador base independiente. Las nuevas instancias son clasificadas sometiéndolas a todos los clasificadores base, donde se combinan sus salidas. Un ventaja importante de esta técnica es que generalmente funciona mejor incluso cuando el clasificador subyacente es capaz de abordar el problema multi-clase directamente [2].

En este trabajo, nos centramos en la estrategia OVO y más específicamente en la fase de combinación de los Sistemas de Múltiples Clasificadores (SMC) formados por clasificadores OVO. Un SMC es un conjunto formado por clasificadores provenientes de diferentes paradigmas de aprendizaje [3]. En el caso de OVO, la idea es que clasificadores diferentes pueden adaptar mejor la clasificación de cada par de clases. Por esta razón, varios trabajos previos han considerado la selección del mejor clasificador para cada par de clases en los SMC [4], [5]. En este trabajo, nuestro objetivo es abordar este problema como un problema de toma de decisión multi-experto, donde tenemos los diferentes expertos (tipos de clasificadores) y sus matrices de confianza para las alternativas consideradas (clases). En este contexto, queremos estudiar la influencia de las funciones de fusión consideradas para combinar las matrices de los diferentes expertos en una única.

En las últimas décadas, el estudio de las funciones de agregación ha crecido significativamente, ya que la necesidad de fusionar o agregar información cuantitativa surge en casi todas las aplicaciones [6], [7], [8], [9]. Sin embargo, en los últimos años, se han propuesto nuevas extensiones de las funciones de agregación, que son capaces de modelar la interacción entre los datos de una mejor manera a pesar de que las propiedades clásicas exigidas a las agregaciones, como la monotonía, no se satisfagan [10], [11]. Desde un punto de vista amplio, estas extensiones se llaman funciones de fusión[12].

Uno de los ejemplos de funciones de fusión que son capaces de modelar la importancia de las entradas o de las interacciones entre ellas son la integral discreta de Choquet [13] y sus extensiones [10], que están basadas en medidas difusas. En este trabajo, proponemos construir estas medidas directamente del conocimiento que podemos extraer de los expertos (clasificadores) utilizando los datos de entrenamiento.

Para realizar este estudio, utilizamos veintiocho conjuntos de datos de KEEL [14] y consideramos el uso de test es-

tadísticos no paramétricos para analizar los resultados obtenidos [15]. Dado que estamos tratando con conjuntos de datos de múltiples clases, no consideraremos solo la precisión para evaluar los resultados, sino que también utilizaremos otras métricas que se centran en la correcta clasificación de todas las clases, como el promedio de precisiones o la media geométrica. Desarrollaremos un estudio jerárquico, donde consideraremos comparaciones intra e inter-familiares para analizar el uso de las diferentes funciones de fusión.

La estructura del artículo es la siguiente. En la sección 2, se recuerdan las diferentes funciones de fusión consideradas en este trabajo. La sección 3 contiene una introducción a la descomposición de los problemas multi-clase, la estrategia OVO y el SMC formado por clasificadores OVO. En la sección 4, describimos con detalle el marco experimental considerado en este estudio, incluyendo como establecer los parámetros de las funciones de fusión parametrizables. La sección 5 contiene el análisis de los resultados obtenidos. Finalmente, en la Sección 6 mostramos las conclusiones obtenidas del estudio.

II. Funciones de fusión

En la literatura reciente, la agregación de información cuantitativa se ha abordado mediante el uso de las funciones de agregación. Una función de agregación se define como una función $f : [0,1]^n \to [0,1]$ (el intervalo $[0,1]$ puede ser extendido a cualquier otro intervalo) tal que $f(0,\ldots,0) = 0$, $f(1,\ldots,1) = 1$, satisfaciendo la propiedad de monotonía, es decir, si $x_i \leq y_i$ para todo $i \in \{1,\ldots,n\}$, entonces $f(x_1,\ldots,x_n) \leq f(y_1,\ldots,y_n)$ [6], [7], [8], [9]. De acuerdo a [6], [7], las principales clases de funciones de agregación son las siguientes: promedios (o medias), conjuntivas, disyuntivas y mixtas. En este trabajo nos hemos centrado principalmente (pero no exclusivamente) en las funciones de agregación promedio, aquellas que están acotadas por el mínimo y el máximo de las entradas.

Sin embargo, en los últimos dos años la propiedad de monotonía de las funciones de agregación se ha visto innecesaria en algunas aplicaciones e incluso se ha generalizado a nuevos tipos de monotonía (ver por ejemplo [12]). A partir de estos estudios, se han definido nuevos conceptos como el de función de pre-agregación [10] o función de fusión interna [11]. Dado que en este artículo modelamos la agregación de datos desde un amplio punto de vista y utilizamos varias funciones no monótonas, hemos utilizado la definición más general de funciones de fusión (ver [12]).

Para clasificar el gran número de funciones de fusión consideradas en este trabajo, hemos establecido una clasificación basada en la necesidad de definir pesos o medidas asociadas a ellas. Básicamente hemos considerado: funciones de fusión no ponderadas, funciones de fusión ponderadas y funciones de fusión basadas en medidas.

Funciones de fusión no ponderadas En esta sub-sección consideramos varias funciones de agregación clásicas:

- La media aritmética $AM(x_1,\ldots,x_n) = \frac{1}{n}(x_1,\ldots,x_n)$;
- La mediana

$$MED(x_1,\ldots,x_n) = \begin{cases} \frac{1}{2}\left(x_{(k)} + x_{(k+1)}\right) & \text{si } n = 2k \text{ es par,} \\ x_{(k)} & \text{si } n = 2k-1 \text{ es impar,} \end{cases}$$

  donde $x_{(k)}$ es el k elemento más largo (más pequeño) de $x_1,\ldots,x_n$;
- La media geométrica $GM(x_1,\ldots,x_n) = (\prod_{i=1}^n x_i)^{\frac{1}{n}}$;
- La media armónica $HM(x_1,\ldots,x_n) = n\left(\sum_{i=1}^n \frac{1}{x_i}\right)^{-1}$.

Funciones de fusión ponderadas En esta sub-sección consideramos funciones de agregación cuyo comportamiento es modelado por un vector de pesos. Esto quiere decir que no todas las entradas son igualmente importantes para calcular el valor agregado, un hecho que claramente nos permite incorporar cierta información externa para el proceso de fusión. Consideramos los vectores de pesos $w = (w_1,\ldots,w_n)$ que satisfagan $w_i \in [0,1]$ y $\sum_{i=1}^n w_i = 1$ [6], [7].

Las funciones de agregación ponderadas son:

- Media aritmética ponderada $WAM(x_1,\ldots,x_n) = \sum_{i=1}^n w_i x_i$;
- Operador OWA $OWA(x_1,\ldots,x_n) = \sum_{i=1}^n w_i x_{(i)}$, donde $(.)$ es una permutación tal que $x_{(1)} \geq \cdots \geq x_{(n)}$.

Funciones de fusión basadas en medidas En esta sub-sección consideramos el conjunto de funciones de fusión basadas en medidas difusas. A diferencia de las funciones de fusión ponderadas, las cuales te permiten modelar la importancia de cada entrada individual, el uso de las medidas difusas nos permiten modelar de manera más general la interacción entre las entradas. En este sentido, la importancia no solo se da a cada entrada individual sino que se asigna también a colecciones (grupos o coaliciones) de entradas. Obviamente, la construcción de la medida difusa es el punto clave para esta familia de funciones de fusión.

Definition 1: Sea $\mathcal{N} = \{1,\ldots,n\}$. Una medida difusa discreta es una función $m : 2^{\mathcal{N}} \to [0,1]$ monótona, es decir, $m(S) \leq m(T)$ siempre que $S \subseteq T$ y satisfaga $m(\emptyset) = 0$ y $m(\mathcal{N}) = 1$.

Una vez visto el concepto de medida difusa podemos definir la integral de Choquet, que es un ejemplo destacado de operador promedio basado en medidas. Empezamos considerando una permutación σ tal que $x_{\sigma(1)} \leq \cdots \leq x_{\sigma(n)}$ con la convención $x_{\sigma(0)} = 0$:

- La integral discreta de Choquet

$$Ch(x_1,\ldots,x_n) = \sum_{i=1}^n (x_{\sigma(i)} - x_{\sigma(i-1)}) * m(\{\sigma(i),\ldots,\sigma(n)\})$$

Como hemos mencionado antes, en [10] se proponen extensiones de las agregaciones, llamados funciones de pre-agregación. Uno de los métodos mas sencillos para construir las pre-agregaciones es cambiando ciertas operaciones en la integral de Choquet. Hemos considerado las siguientes funciones de pre-agregación:

- Integral de Choquet basada en la t-norma mínimo

$$Ch_M(x_1,\ldots,x_n) = \sum_{i=1}^n \text{mín}\{x_{\sigma(i)} - x_{\sigma(i-1)}, m(\{\sigma(i),\ldots,\sigma(n)\})\};$$

- Integral de Choquet basada en la t-norma de Lukasiewicz

$$Ch_L(x_1,\ldots,x_n)=\sum_{i=1}^{n} \text{máx}\{0, x_{\sigma(i)}-x_{\sigma(i-1)}+m(\{\sigma(i),\ldots,\sigma(n)\})-1\}$$

III. One-vs-One para problemas multi-clase y sistemas de múltiples clasificadores

En esta sección introducimos los problemas de clasificación y, más específicamente, la estrategia One-vs-One (OVO) para tratar los problemas de clasificación multi-clase y los sistemas de múltiples clasificadores con el objetivo de mejorar el rendimiento de la clasificación mediante la combinación de varios clasificadores.

En aprendizaje automático, un problema de clasificación consiste en aprender un sistema (clasificador) capaz de predecir la salida deseada (etiqueta) para cada patrón de entrada. Formalmente, el objetivo es buscar un función $\mathbb{A}^i \rightarrow \mathbb{C}$ donde $a_1,\ldots,a_i \in \mathbb{A}$ son los atributos que caracterizan cada ejemplo de entrada $x_1,\ldots,x_n$ y cada entrada tiene asociada la salida deseada $y_j \in \mathbb{C} = \{c_1,\ldots,c_m\}$. Se espera que el clasificador generalice bien a nuevos ejemplos del problema que no se han considerado en el entrenamiento, esto es, debería tener una buena habilidad de generalización.

Un problema de clasificación multi-clase se da cuando el número de clases es mayor que dos ($|\mathbb{C}| > 2$). Estos problemas se consideran mas difíciles que los problemas de clasificación binarios dado que las fronteras de decisión son generalmente más complejas y existe un mayor solapamiento entre clases. Es por esto que se crearon las estrategias de descomposición [1], para tratar con los problemas de clasificación multi-clase dividiendo el problema original en problemas de clasificación binarios más fáciles de resolver. Por lo tanto, se aprende un clasificador binario por cada nuevo problema, conocidos como clasificadores base, y se combinan las salidas de estos clasificadores cuando se quiere clasificar un nuevo ejemplo no etiquetado. Se ha probado que estas estrategias no son solo útiles cuando se trabaja con clasificadores que solo soportan problemas binarios (como las maquinas de vectores de soporte, SVM [16]), sino que también con clasificadores que soportan la clasificación multi-clase. En estos casos, el rendimiento final se puede mejorar descomponiendo el problema [2].

III-A. La estrategia One-Vs-One

La estrategia OVO es la estrategia de descomposición más utilizadas. En OVO, se divide un problema con m clases en tantos problemas como posibles pares de clases haya, generando $m(m-1)/2$ sub-problemas que son abordados mediante clasificadores base independientes. En cada sub-problema, solo se consideran los ejemplos que pertenezcan al par de clases considerado, descartando el resto. A la hora de clasificar un nuevo ejemplo, se somete éste a todos los clasificadores cuyas salidas tienen que ser combinadas para decidir la clase final. Para realizar la combinación, generalmente se almacenan todas las salidas en una matriz de confianza (Eq. 1) donde cada posición $r_{ij}, r_{ji} \in [0,1]$ corresponde al grado de confianza del clasificador distinguiendo las clases $\{C_i, C_j\}$. Dado que la mayoría de clasificadores devuelven la confianza basada en estimaciones de probabilidad, generalmente r_{ji} se calcula como $r_{ji} = 1 - r_{ij}$. Sin embargo, si este no es el caso, como ocurre con los clasificadores basados en reglas difusas [17], la matriz de confianza se normaliza para que $r_{ij} + r_{ji} = 1$ [17].

$$R = \begin{pmatrix} - & r_{12} & \cdots & r_{1m} \\ r_{21} & - & \cdots & r_{2m} \\ \vdots & & & \vdots \\ r_{m1} & r_{m2} & \cdots & - \end{pmatrix} \quad (1)$$

Finalmente, se combinan las salidas de los clasificadores base por cada fila (clase) y se asigna la clase que consiga la mayor confianza total. En la literatura, se han desarrollado varias estrategias de combinación para este propósito. Se realizó una completa revisión en [2] y se han desarrollado varias extensiones de combinación considerando la selección de clasificadores y el mecanismo de ponderación [18], [19]. En este trabajo, consideramos la estrategia del voto ponderado (WV) [20] ya que se ha demostrado que es un método simple y robusto. En este método, cada clasificador base vota por ambas clases basándose en la confianza dada por el par de clases. Finalmente, se devuelve la clase con mayor valor

$$Class = arg \max_{i=1,\ldots,m} \sum_{1 \le j \ne i \le m} r_{ij}. \quad (2)$$

III-B. Sistema de Múltiples clasificadores y OVO

La estrategia OVO se puede ver como un modelo ensemble [2], donde se utiliza una combinación de clasificadores con el objetivo de mejorar los resultados de un único clasificador. Este término se considera generalmente para describir la combinación de variantes menores del mismo clasificador. De otra manera, un sistema de múltiples clasificadores (SMC) es una categoría más amplia incluyendo esas combinaciones considerando el uso de diferentes modelos de clasificación [3].

Recientemente, se han considerado varios trabajos centrados en la hibridación de ensembles OVO (donde se utiliza el mismo clasificador base para cada sub-problema, p. ej. SVM) con SMC. Esto es, para construir varios ensembles OVO con diferentes clasificadores (por ejemplo, uno utilizado SVM, otro utilizando métodos de inducción de reglas y otro utilizando árboles de decisión) y para combinar las salidas de todos los ensembles OVO para tomar la decisión final.

Otros trabajos se han centrado en seleccionar estática o dinámicamente el mejor clasificador para distinguir cada par de clases [4], [5]. Sin embargo, nuestro objetivo en este trabajo es ver el problema desde un perspectiva diferente para probar el uso de diferentes funciones de fusión en la combinación de los diferentes clasificadores.

Una vez que se han entrenado todos los clasificadores OVO del SMC (asumiendo que teniendo tres clasificadores diferentes y un problema de cuatro clases, tendríamos $3 \cdot 4 \cdot (4-1)/2$ clasificadores), se clasifica un nuevo ejemplos sometiéndolo a todos los clasificadores. Como resultado, en vez de obtener una única matriz de confianza, tendríamos tantas matrices como clasificadores considerados (tres en nuestro ejemplo). El problema es cómo combinar estas matrices en una única donde podamos aplicar la estrategia WV para clasificar el ejemplo. Es por esto que podemos entender el problema como un problema

de toma de decisión multi-experto. Nuestra propuesta en este trabajo es combinar las diferentes matrices de confianza utilizando las funciones de fusión. Nuestro objetivo es estudiar cómo el uso de diferentes funciones de fusión afecta al rendimiento del SMC. Para ello, consideraremos las diferentes funciones de fusión mencionadas en la sección anterior y propondremos diferentes mecanismos para asignar los pesos o crear las medidas difusas en las funciones que requieran estos parámetros. Más detalles acerca de la obtención de dichos parámetros se dan en la Sección IV-B.

IV. Marco experimental

IV-A. Datasets, evaluación, test estadísticos y algoritmos

Para llevar a cabo el estudio experimental, hemos utilizado veintiocho conjuntos de datos numéricos del repositorio de datos de KEEL [14], cuyas características principales se muestran en la Tabla I.

Tabla I
Resumen de las características de los conjuntos de datos utilizados en el estudio experimental.

Dataset	#Ej.	#Atr.	#Clas.	Dataset	#Ej.	#Atr.	#Clas.
autos	159	25	6	nursery	1296	8	5
balance	625	4	3	pageblocks	548	10	5
car	1728	6	4	penbased	1100	16	10
cleveland	297	13	5	satimage	643	36	7
contraceptive	1473	9	3	segment	2310	19	7
dermatology	358	34	6	shuttle	2175	9	7
ecoli	336	7	8	splice	319	60	3
flare	1066	11	6	tae	151	5	3
glass	214	9	7	thyroid	720	21	3
hayes-roth	132	4	3	vehicle	846	18	4
iris	150	4	3	vowel	990	13	11
led7digit	500	7	10	wine	178	13	3
lymphography	148	18	4	yeast	1484	8	10
newthyroid	215	5	3	zoo	101	16	7

El resultado de cada método y conjunto de datos se ha obtenido utilizando validación cruzada con 5 particiones. Además, para analizar apropiadamente los resultados obtenidos, hemos aplicado test estadísticos no paramétricos[15]. Más específicamente, hemos utilizado el test de Wilcoxon para comparar un par de métodos, mientras que se considera el test de rangos alineados de Friedman para comparar un grupo de métodos con el objetivo de detectar si existen diferencias estadísticas. En tal caso, se utiliza el test *post-hoc* de Holm para buscar los algoritmos que rechazan la hipótesis nula de equivalencia frente al método de control seleccionado.

Dado que estamos tratando con problemas multi-clase, hemos considerado tres medidas de rendimiento para analizar los resultados obtenidos: el ratio de precisión (Acc), esto es, el ratio de los ejemplos clasificados correctamente; media aritmética (AvgAcc) y media geométrica (GM) de los ratios de los ejemplos correctamente clasificados por cada clase. Por lo tanto, Acc nos da un medida global de la calidad del algoritmo, mientras que AvgAcc y GM se centran más en medir apropiadamente si todas las clases del problema se están clasificando apropiadamente.

Respecto a los algoritmos de clasificación considerados para formar nuestro SMC, hemos considerado los siguientes (los cuales también fueron considerados en nuestros trabajos previos [2], [18], [19]): *Support Vector Machine* (SVM) [16], *C4.5 decision tree* [21], *k−Nearest Neighbors* (*k*NN) [22], *Repeated Incremental Pruning to Produce Error Reduction* (Ripper) [23], *Positive Definite Fuzzy Classifier* (PDFC)[24].

Estos clasificadores se han entrenado utilizando los parámetros mostrados en la Tabla II. Estos valores son comunes para todos los problemas, y se han seleccionado de acuerdo a las recomendaciones de los autores correspondientes, que son sus valores por defecto incluidos en KEEL, software [14] utilizado para realizar nuestros experimentos.

Tabla II
Especificación de los parámetros para los clasificadores base utilizados en la experimentación.

Algoritmo	*Parámetros*
SVM_{Poly}	C = 1.0, Tolerance Parameter = 0.001, Epsilon = 1.0E-12, Kernel Type = Polynomial Polynomial Degree = 1, Fit Logistic Models = True
SVM_{Puk}	C = 100.0, Tolerance Parameter = 0.001, Epsilon = 1.0E-12, Kernel Type = Puk PukKernel ω = 1.0, PukKernel σ = 1.0, Fit Logistic Models = True
C4.5	Prune = True, Confidence level = 0.2, Minimum number of item-sets per leaf = 2
3NN	$k = 3$, Distance metric = HVDM
Ripper	Size of growing subset = 66 %, Repetitions of the optimization stage = 2
PDFC	C = 100.0, Tolerance Parameter = 0.001, Epsilon = 1.0E-12, Kernel Type = Polynomial Polynomial Degree = 1, PDRF Type = Gaussian

Debemos recordar que las matrices de confianza representan las confianzas obtenida de los clasificadores. Dado que no todos los clasificadores devuelven la confianza directamente, detallamos cómo se han obtenido.

- **SVM –** Estimación de la probabilidad de la SVM
- **C4.5 –** Precisión de la hoja realizando la predicción (ejemplos de entrenamiento bien clasificados divido por el número total de ejemplos de entrenamiento cubiertos).
- ***k*NN –** Confianza basada en distancia.$Confianza = \frac{\sum_{l=1}^{k} \frac{e_l}{d_l}}{\sum_{l=1}^{k} \frac{1}{d_l}}$ Donde d_l es la distancia entre el patrón de entrada y el vecino l y $e_l = 1$ si el vecino l es de la clase y 0 en otro caso.
- **Ripper –** Precisión de la regla utilizada en la predicción (como en C4.5 considerando reglas en vez de hojas).
- **PDFC –** La predicción del clasificador, esto es, la confianza es 1 para la clase predicha.

IV-B. Parámetros para las funciones de fusión

En lo sucesivo presentamos el método para estimar los parámetros requeridos por algunas funciones de fusión.

Cálculo de pesos Para la media aritmética ponderada necesitamos establecer los pesos para cada entrada (clasificador, p. ej., SVM, 3NN, ...). Establecemos cada peso como la precisión normalizada de cada método en el conjunto de entrenamiento, esto es, $w_i = \frac{Acc_i}{\sum_{j=1}^{n} Acc_j}$ para todo $i \in \{1, \ldots, n\}$.

Además, hemos utilizado dos versiones diferentes para las funciones de fusión ponderadas: un enfoque global y otro local. En el enfoque global, asignamos un peso a cada clasificador. Sin embargo, en el enfoque local, cada clasificador obtiene un peso por cada problema individual (precisión sobre cada par de clases).

El cálculo de los pesos para los operadores OWA se realiza mediante el uso de cuantificadores difusos crecientes (ver [25]), y son dados por $w_i = Q\left(\frac{i}{n}\right) - Q\left(\frac{i-1}{n}\right)$ para

todo $i \in \{1,\ldots,n\}$. En este trabajo hemos considerado 3 cuantificadores difusos diferentes obteniendo tres operadores OWA: 'al menos la mitad' (OWA_alh) con $a = 0, b = 0{,}5$; 'la mayor cantidad posible' (OWA_amap) con $a = 0{,}5, b = 1$; y 'la mayoría de' (OWA_mot) con $a = 0{,}3, b = 0{,}8$.

Valores de la medida difusa Para las funciones de fusión basadas en medidas, necesitamos construir una medida difusa $m : 2^{\mathscr{N}} \to [0,1]$ con $\mathscr{N} = \{1,\ldots,n\}$, siendo n el número de clasificadores considerados. Empezamos considerando la medida difusa uniforme m_U la cual se construye con $m_U(A) = \frac{|A|}{n}$ para todo $A \subseteq \mathscr{N}$ (la integral de Choquet con esta medida equivale a la media aritmética).

Sin embargo, para capturar las interacciones entre clasificadores, utilizaremos no solo la precisión de los clasificadores individuales sino también la precisión de cada posible combinación de clasificadores. Denotaremos estas precisiones como Acc_A, para todo $A \subseteq \mathscr{N}$. Ahora, por cada nivel de la medida difusa (todos los elementos de la medida difusa con la misma cardinalidad), calculamos la media aritmética de las precisiones en cada nivel correspondiente, llamándola $MeanAcc_i$ para todo $i \in \{1,\ldots,n\}$. Finalmente, el valor de la medida difusa para cada $A \subseteq \mathscr{N}$ vendrá dado por

$$m(A) = m_U(A)(1 + Acc_A - MeanAcc_{|A|}). \quad (3)$$

Analizando esta expresión, el valor de la medida difusa asociado a los clasificadores que son mejores que la precisión media en el mismo nivel aumentarán (respecto a la medida uniforme) y el valor de aquellos que son peores decrementarán. De manera similar al cálculo anterior de pesos, consideraremos un enfoque global y otro local.

Es importante hacer notar que no podemos garantizar la monotonía de m para todo posible valor de Acc y $MeanAcc$. Para corregir esto, y basándonos en la verificación de monotonía dada en [26], hemos utilizado una corrección top-down: empezamos en el nivel superior de la medida $m(\mathscr{N})$ y vamos evaluando los valores de la medida en los niveles inferiores $m(A)$ donde $|A| = n-1$. Si encontramos algún A tal que $m(A) > m(\mathscr{N})$, entonces establecemos $m(A) = m(\mathscr{N})$. Una vez que el nivel $n-1$ es verificado (con respecto al nivel n), verificamos el nivel $n-2$ con respecto al nivel $n-1$. Repetimos este proceso hasta que la medida satisfaga el criterio de monotonía.

V. Estudio experimental

Por un lado, la Tabla III muestra las precisiones (Acc), la media aritmética (AvgAcc) y la media geométrica (GM) de las precisiones de cada clase obtenidas sobre el conjunto de test utilizando diferentes funciones de fusión para combinar las matrices OVO en el SMC. El mejor resultado de cada métrica esta subrayado.

Por otro lado, la Figura 1 resume el estudio estadístico llevado a cabo para cada métrica de rendimiento para analizar cuál es la función de fusión que mejor funciona en cada caso. Para crear esta figura, hemos enfrentada las funciones dentro de cada familia utilizando el test de rangos alineados de Friedman. Luego, se comparan los mejores de cada familia en una etapa final que nos da la mejor función de fusión. En cada comparación, mostramos los rangos obtenidos por cada método (cuanto menor, mejor) y marcamos con **negrita** los rangos en los que el test post-hoc detecta diferencias significativas (con $\alpha = 0{,}1$) en favor del método ganador.

Tabla III
Resultados medios obtenidos en todos los conjuntos de test con diferentes funciones de fusión para cada métrica de rendimiento

Family	Fusion	Acc	AvgAcc	GM
Unweighted	AM	0.8544	0.7911	0.6240
	MED	0.8580	0.7951	0.6332
	GM	0.8285	0.7535	0.5588
	HM	0.8252	0.7515	0.5610
Weighted	WAM	0.8544	0.7916	0.6308
	WAM_local	0.8481	0.7893	0.6344
	OWA_alh	0.8573	0.7996	0.6448
	OWA_amap	0.8496	0.7815	0.6073
	OWA_mot	0.8554	0.7921	0.6254
Choquet	Ch	0.8552	0.7940	0.6305
	Ch_local	0.8541	0.7924	0.6334
	Ch_L	0.8487	0.7789	0.6087
	Ch_L_local	0.8502	0.7803	0.6088
	Ch_M	0.8548	0.7939	0.6395
	Ch_M_local	0.8556	0.7964	0.6397

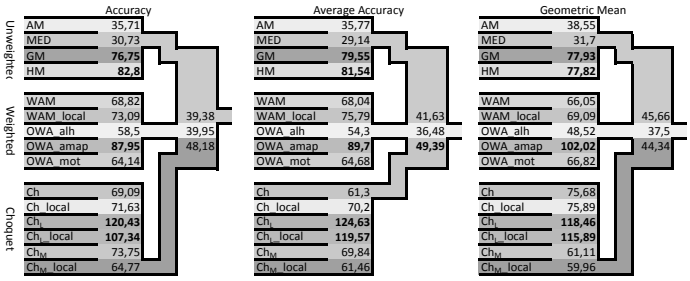

Figura 1. Estudio estadístico jerárquico comparando las funciones de fusión en cada familia y la mejor de cada familia para cada medida de rendimiento utilizando el test de rangos alineados de Friedman.

Finalmente, completamos el análisis estadístico comparando la media aritmética (AM, función comúnmente utilizada) con el ganador de cada familia. Estas comparaciones se muestran en la Tabla IV, donde se muestra el p-valor de la comparación y si existen o no diferencias estadísticas en **negrita**.

Tabla IV
Comparación de la AM contra las mejores funciones de cada familia utilizando el test de Wilcoxon.

Perf. Measure	Unweighted	Weighted	Choquet
Acc	MED	OWA_alh	Ch_M_local
	0.0152	**0.0298**	0.7610
AvgAcc	MED	OWA_alh	Ch
	0.0194	**0.0126**	**0.0994**
GM	MED	OWA_alh	Ch_M_local
	0.0169	**0.0036**	**0.0400**

Viendo estos resultados, podemos observar los siguientes hechos:

- Analizando los resultados por cada familia, vemos que dentro de las funciones no ponderadas, AM y MED son

las que mejores resultados obtienen. Viendo los test de Wilcoxon vemos que MED supera estadísticamente a AM en las tres las medidas de rendimiento.
Analizando las funciones ponderadas, OWA_alh es la que mejor funciona, aunque solo existen diferencias estadísticas con respecto a OWA_amap. Esto es posible debido a que dicho OWA actúa como el promedio de los tres clasificadores más competitivos. En este caso, obtener los pesos de los datos (WAM y su versión local) obtiene peores resultados que estableciendo los pesos de manera predefinida. Finalmente, en cuanto a las funciones basadas en medidas difusas, las pre-agregaciones que utilizan el mínimo son mejores en casi todos los casos, mostrando robustez frente a la medida de rendimiento considerada (aunque no se encuentran diferencias estadísticas)
Se podrían esperar mejores resultados en los casos donde los parámetros se obtienen de los datos. Aunque no se han encontrado diferencias estadísticas con respecto a la WAM y a la Choquet, en el futuro nuestro objetivo es centrarnos en estas funciones e intentar mejorar el modelado de los parámetros para ser más competitivos. De hecho, los operadores OWA son un caso particular de la integral de Choquet y, por lo tanto, parece razonable poder obtener una medida difusa que por lo menos llegue al comportamiento de cualquier OWA.

- Finalmente, analizando la Tabla IV se puede ver que la función comúnmente utilizada (AM) es superada estadísticamente por la MED y la OWA_alh en todos los casos y por la Choquet en los casos de AvgAcc y GM. Por lo tanto, existe un margen de mejora considerando diferente funciones de fusión.

VI. Conclusiones

En este trabajo, hemos considerado un SMC formado por clasificadores OVO y hemos enfocado la fase de combinación como un problema de toma de decisión multi-experto. En consecuencia, hemos desarrollado un estudio empírico completo para analizar el comportamiento de las diferentes funciones de fusión. También hemos propuesto diferentes formas de obtener los parámetros para las funciones ponderadas y basadas en medias difusas utilizando los datos. Aunque se esperarían mejores resultados para ese tipo de funciones, los operadores OWA son los que mejores resultados obtienen. Dado que los OWA son un caso particular de medida difusa, se motiva un estudio para construir las medidas difusas de diferentes maneras para mejorar la calidad de sus resultados.

Agradecimientos.: Este trabajo ha sido apoyado en parte por el Ministerio Español de Ciencia y Tecnología bajo el Proyecto TIN2016-77356-P (AEI/FEDER, UE).

Referencias

[1] A. C. Lorena, A. C. Carvalho, and J. M. Gama, "A review on the combination of binary classifiers in multiclass problems," *Artificial Intelligence Review*, vol. 30, no. 1-4, pp. 19–37, 2008.

[2] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognition*, vol. 44, no. 8, pp. 1761 – 1776, 2011.

[3] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 66–75, 1994.

[4] S. Kang, S. Cho, and P. Kang, "Multi-class classification via heterogeneous ensemble of one-class classifiers," *Engineering Applications of Artificial Intelligence*, vol. 43, pp. 35–43, 2015.

[5] I. Mendialdua, J. M. Martínez-Otzeta, I. Rodriguez-Rodriguez, T. Ruiz-Vazquez, and B. Sierra, "Dynamic selection of the best base classifier in One versus One," *Knowledge-Based Systems*, vol. 85, pp. 298–306, 2015.

[6] G. Beliakov, A. Pradera, and T. Calvo, *Aggregation Functions: A Guide for Practitioners*. Springer, 2007.

[7] G. Beliakov, H. Bustince, and A. Pradera, *A Practical Guide to Averaging Functions*, 2nd ed. Springer, 2015.

[8] T. Calvo, G. Mayor, and R. Mesiar, *Aggregation Operators. New Trends and Applications*. Physica-Verlag, 2002.

[9] M. Grabisch, J. L. Marichal, R. Mesiar, and E. Pap, *Aggregation Functions*. Cambridge University Press, 2009.

[10] G. Lucca, J. Sanz, G. Dimuro, B. Bedregal, R. Mesiar, A. Kolesárová, and H. Bustince, "Preaggregation functions: Construction and an application," *IEEE Transactions on Fuzzy Systems*, vol. 24, pp. 260–272, 2016.

[11] D. Paternain, M. J. Campión, H. Bustince, I. Perfilieva, and R. Mesiar, "Internal fusion functions," *IEEE Transactions on Fuzzy Systems*, InPress.

[12] H. Bustince, J. Fernandez, A. Kolesárová, and R. Mesiar, "Directional monotonicity of fusion functions," *European Journal of Operational Research*, vol. 244, pp. 300–308, 2015.

[13] G. Choquet, "Theory of capacities," *Ann. Inst. Fourier*, vol. 5, pp. 1953–1954, 1953.

[14] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17:2-3, pp. 255–287, 2011.

[15] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Information Sciences*, vol. 180, pp. 2044–2064, 2010.

[16] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[17] M. Elkano, M. Galar, J. Sanz, A. Fernandez, E. Barrenechea, F. Herrera, and H. Bustince, "Enhancing multi-class classification in farc-hd fuzzy classifier: On the synergy between n-dimensional overlap functions and decomposition strategies," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 5, pp. 1562 – 1580, 2015.

[18] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "Dynamic classifier selection for one-vs-one strategy: Avoiding non-competent classifiers," *Pattern Recognition*, vol. 46, no. 12, pp. 3412–3424, 2013.

[19] M. Galar, A. Fernández, E. Barrenechea, and F. Herrera, "DRCW-OVO: Distance-based relative competence weighting combination for one-vs-one strategy in multi-class problems," *Pattern Recognition*, vol. 48, no. 1, pp. 28–42, 2015.

[20] E. Hüllermeier and S. Vanderlooy, "Combining predictions in pairwise classification: An optimal adaptive voting strategy and its relation to weighted voting," *Pattern Recognition*, vol. 43, no. 1, pp. 128–142, 2010.

[21] J. R. Quinlan, *C4.5: Programs for Machine Learning*, 1st ed. San Mateo-California: Morgan Kaufmann Publishers, 1993.

[22] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, pp. 37–66, 1991.

[23] W. W. Cohen, "Fast effective rule induction," in *ICML'95: Proc. of the Twelfth Int. Conf. on Machine Learning*, 1995, pp. 1–10.

[24] Y. Chen and J. Z. Wang, "Support vector learning for fuzzy rule-based classification systems," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 6, pp. 716–728, 2003.

[25] R. Yager, "Quantifier guided aggregation using owa operators," *International Journal of Intelligent Systems*, vol. 11, pp. 49–73, 1998.

[26] M. Grabisch, "A new algorithm for identifying fuzzy measures and its application to pattern recognition," in *Int. Joint Conf. of the 4th IEEE Int. Conf. on Fuzzy Systems and the 2nd Int. Fuzzy Engineering Symposium*, 1995, pp. 145–150.