XIX Congreso Español sobre Tecnologías y Lógica Fuzzy (XIX ESTYLF)

ESTYLF 5: Sesión Especial: Soft Computing en Aprendizaje

Organizadores:

Mikel Galar, José Antonio Sanz

A PageRank-based method to extract fuzzy expressions as features in supervised classification problems*

*Note: The full contents of this paper have been published in the volume *Lecture Notes in Artificial Intelligence 11160* (LNAI 11160)

Pablo Carmona
Industrial Engineering School
University of Extremadura
Badajoz, Spain

Juan Luis Castro
Department of Computer Science and Artificial Intelligence
University of Granada
Granada, Spain

Jesús Lozano
Industrial Engineering School
University of Extremadura
Badajoz, Spain

José Ignacio Suárez
Industrial Engineering School
University of Extremadura
Badajoz, Spain

***Abstract*—This work presents a new ranking method inspired on PageRank to reduce the dimensionality of the feature space in supervised classification problems. More precisely, as it relies on a weighted directed graph, it is ultimately inspired on TextRank, a PageRank based method that adds weights to the edges to express the strength of the connections between nodes. The method is based on dividing each original feature used to describe the training set into a set of fuzzy predicates and then ranking all of them by their ability to differentiate among classes in the light of this training set. The fuzzy predicates with the best scores can be then used as new features, replacing the original ones. The novelty of the proposal relies on being an approach halfway between feature selection and feature extraction approaches, being able to improve the discrimination ability of the original features but preserving the interpretability of the new features in the sense that they are fuzzy expressions. Preliminary results supports the suitability of the proposal.**

***Index Terms*—fuzzy logic, supervised classification, ranking methods, feature selection, feature extraction, PageRank, TextRank**

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

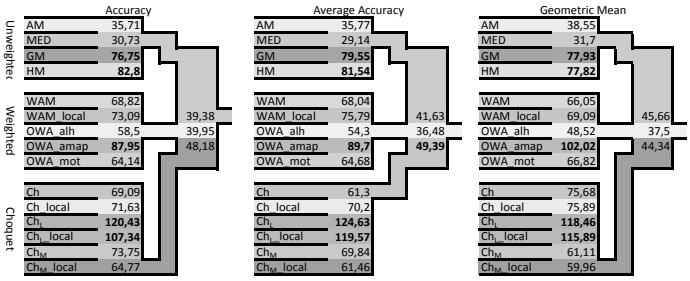

Figura 1. Estudio estadístico jerárquico comparando las funciones de fusión en cada familia y la mejor de cada familia para cada medida de rendimiento utilizando el test de rangos alineados de Friedman.

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.

Clustering difuso con pertenencias intervalares

Aranzazu Jurio
Departamento de Estadística,
*I*nformática y Matemáticas
*I*nstituto de Smart Cities
Universidad Pública de Navarra
Pamplona, España
aranzazu.jurio@unavarra.es

Humberto Bustince
Departamento de Estadística,
*I*nformática y Matemáticas
*I*nstituto de Smart Cities
Universidad Pública de Navarra
Pamplona, España
bustince@unavarra.es

Vicenç Torra
School of Informatics
University of Skövde
Skövde, Sweden
vtorra@ieee.org

***Resumen*—En este trabajo estudiamos cómo solucionar el problema que presentan los outliers a la hora de realizar un proceso de agrupamiento. Para ello, presentamos una función objetivo que extiende a la del algoritmo Fuzzy Cluster Means, mediante el uso de conjuntos intervalo-valorados difusos. En este caso, las pertenencias de cada dato a cada grupo se representan mediante intervalos. Posteriormente, presentamos un algoritmo que minimiza la función objetivo propuesta y mostramos cómo se comporta ante diferentes conjuntos de datos.**

***Index Terms*—clustering, intervalo, pertenencia**

I. Introducción

El problema de clustering o agrupamiento es un problema de clasificación no supervisada cuyo objetivo es encontrar los grupos naturales existentes en un conjunto de datos. Para ello, se basa en la idea de que los datos que pertenecen a un mismo grupo deben compartir características similares, mientras que los datos que pertenecen a diferentes grupos deben diferenciarse en dichas características [4].

Los algoritmos de agrupamiento se pueden dividir de manera general en dos tipos: algoritmos jerárquicos y algoritmos de particiones. Los algoritmos jerárquicos crean un árbol (dendograma) que mide las similitudes entre los datos [5] [9]. Por su parte, los algoritmos de particiones separan los datos en un número prefijado de grupos. Cada uno de esos grupos se representa mediante un centroide, que es el punto cuya suma de distancias desde todos los datos del grupo a sí mismo es mínima [6] [7] [8]. En este trabajo nos centramos en los algoritmos de particiones.

Dentro de los algoritmos de particiones uno de los más conocidos y utilizados es el k-means [3] [8]. En este algoritmo se separan todos los datos en c grupos y se calculan los centroides de cada grupo. El objetivo es minimizar la suma de las distancias de cada dato a su centroide correspondiente.

$$J = \sum_{i=1}^{c} \sum_{x_k \in cluster_i} ||x_k - v_i||_A^2$$

Este trabajo ha sido parcialmente financiado por el Ministerio de Educación, Cultura y Deporte mediante el programa José Castillejo para estancias de movilidad en el extranjero de jóvenes doctores. También ha sido parcialmente financiado por el Ministerio de Economía, Industria y Competitividad del Gobierno de España mediante el proyecto TIN2016-77356-P (AEI/FEDER, UE).

donde $||x||_A = \sqrt{x^t A x}$ es cualquier norma asociada a un producto escalar.

Uno de los problemas que presenta el algoritmo k-means, así como muchos algoritmos de agrupamiento de particiones, es que no son capaces de manejar situaciones en las que los grupos de datos se encuentran solapados. En esos casos, los datos que se encuentran en la zona solapada entre dos o más grupos deberían pertenecer a todos esos grupos y no solo a uno de ellos.

Una de las maneras de solucionar este problema es utilizando la lógica difusa [10]. De este modo, cada uno de los datos puede pertenecer a más de un grupo, con diferente valor de pertenencia. Estos valores de pertenencia son números entre 0 y 1. El algoritmo Fuzzy Cluster Means (FCM) [1] extiende la idea del algoritmo k-means empleando la lógica difusa.

En el FCM, cada dato tiene una pertenencia total de 1, que reparte entre todos los grupos. De esta manera, un mismo dato puede pertenecer a todos los grupos existentes, siempre y cuando la suma de sus valores de pertenencia sea 1. El objetivo es minimizar la suma ponderada de las distancias de cada dato a todos los centroides. Los pesos de esta suma vienen dados por el valor de pertenencia de cada dato a cada grupo.

$$J = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m ||x_k - v_i||_A^2$$

Este algoritmo consigue solucionar el problema del solapamiento entre grupos. Sin embargo, cuando existen outliers entre los datos a clasificar, el FCM no es capaz de detectarlos y por ello sus resultados se ven distorsionados.

Para solucionar este problema, en este trabajo presentamos un nuevo algoritmo de agrupamiento que extiende el FCM, de tal forma que es capaz de detectar los datos que no pertenecen a los grupos naturales existentes en el conjunto de datos.

De la misma forma que el uso de los conjuntos difusos permite aportar nueva información al proceso de agrupamiento, en nuestra propuesta utilizamos una extensión de los conjuntos difusos: los conjuntos intervalo-valorados difusos. En este trabajo, utilizamos dichos conjuntos para cuantificar las pertenencias, por lo que cada dato va a pertenecer a todos los grupos con un valor de pertenencia que es un intervalo en [0,1]. Utilizamos la amplitud de dicho intervalo

para representar la seguridad que tenemos de que ese dato pertenezca a los grupos presentes en el dataset que estamos clasificando. De esta forma, si estamos totalmente seguros de que un dato pertenece a uno o varios de los grupos existentes, las pertenencias de dichos datos a todos los grupos tendrán amplitud 0. Por el contrario, si estamos totalmente seguros de que un dato no pertenece a ninguno de los grupos presentes, las pertenencias de ese dato a todos los grupos tendrán amplitud 1, que es el máximo permitido.

El resto de este trabajo está organizado de la siguiente forma: en la Sección II repasamos brevemente el algoritmo Fuzzy Cluster Means; en la Sección III explicamos en detalle el algoritmo que proponemos en este trabajo y en la Sección IV vemos cómo se comporta mediante el uso de varios ejemplos de datos. Finalmente, en la Sección V mostramos las conclusiones obtenidas.

II. FCM

El Fuzzy Cluster Means (FCM) [1] es uno de los algoritmos de agrupamiento difuso más utilizados. Al utilizar conjuntos difusos, permite que cada uno de los datos pertenezca a más de un grupo al mismo tiempo. De hecho, basa su idea en que cada dato debe pertenecer a todos los grupos presentes con un grado de pertenencia dado. Estos grados de pertenencia son valores entre 0 y 1, de tal forma que la suma de los valores de pertenencia de cada dato a todos los grupos siempre sea 1.

Con esta premisa, el algoritmo FCM trata de minimizar la suma de distancias ponderadas de cada dato a todos los centroides, haciendo que los pesos de ponderación sean proporcionales al valor de pertenencia del dato a ese grupo.

$$J = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m ||x_k - v_i||_A^2$$

donde x_k es el k-ésimo dato a clasificar, v_i es el centroide del grupo i, u_{ik} es el grado de pertenencia del dato k al grupo i y m es un valor real mayor que 1. Además, se deben cumplir tres restricciones:

- $u_{ik} \geq 0, \quad k = 1..n,\ i = 1..c$
- $\sum_{k=1}^{n} u_{ik} > 0, \quad i = 1..c$
- $\sum i = 1^c u_{ik} = 1, \quad k = 1..n$

La solución a este problema es un proceso iterativo que comienza con unos centroides elegidos aleatoriamente. A partir de los centroides se pueden calcular los nuevos valores de pertenencia de cada dato a cada grupo:

$$u_{ik} = \left(\sum_{j=1}^{c} \left(\frac{||x_k - v_i||_A}{||x_k - v_j||_A} \right)^{2/(m-1)} \right)^{-1}$$

$k = 1..n$, $i = 1..c$. Y a partir de los valores de pertenencia se pueden calcular los nuevos centroides:

$$v_i = \frac{\sum_{k=1}^{n}(u_{ik})^m x_k}{\sum_{k=1}^{n}(u_{ik})^m}$$

$i = 1..c$. El proceso termina cuando los cambios en los valores son suficientemente pequeños.

Uno de los problemas que presenta el algoritmo FCM se produce cuando el conjunto de datos a clasificar presenta ruido, o outliers. En estos casos, todos los datos se asignan a los diferentes grupos, por lo que los datos alejados pueden modificar erróneamente sus centroides. En la Figura 1 podemos ver un conjunto de datos marcados con asteriscos negros. Claramente podemos ver dos grupos de datos que se solapan y un dato que no pertenece a ninguno de los grupos. Marcados con círculos rojos se ven los centroides de los dos grupos detectados por el algoritmo FCM.

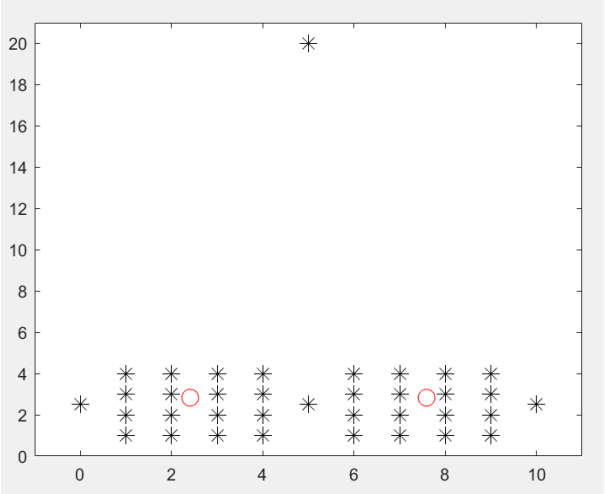

Figura 1. Ejecución del algoritmo FCM sobre un conjunto de datos con un outlier. Conjunto de datos a clasificar marcados con asteriscos negros. Centroides obtenidos por el FCM marcados con círculos rojos.

Los dos grupos de datos están centrados verticalmente en el punto 2.5. Sin embargo, como el dato situado en el punto $(5, 20)$ pertenece a ambos grupos, los centroides se desplazan hacia arriba hasta el valor vertical 2.83.

Además, si analizamos los valores de pertenencia de cada dato a los dos grupos, vemos que el dato situado en el punto (5, 2.5) tiene un valor de pertenencia de 0.5 a cada uno de los grupos. Por su parte, el dato situado en el punto (5, 20) también tiene un valor de pertenencia de 0.5 a cada grupo. Por tanto, el algoritmo nos indica que ambos datos son iguales a la hora de formar los dos grupos. Sin embargo, observando la figura, vemos que uno de los dos datos pertenece a la zona solapada de los dos grupos mientras que el otro dato no pertenece a ninguno de los grupos.

A partir de este ejemplo, podemos afirmar que el algoritmo Fuzzy Cluster Means no funciona adecuadamente cuando en el conjunto de datos a clasificar hay datos que no pertenecen a ninguno de los grupos existentes.

III. Nuevo algoritmo de clustering intervalar

En esta sección explicamos nuestra propuesta de algoritmo de agrupamiento. Su principal novedad es que hace uso de los conjuntos intervalo-valorados difusos para representar las pertenencias de los datos a cada cluster.

Llamamos $L([0,1])$ al conjunto de todos los subintervalos cerrados en $[0,1]$, es decir,

$$L([0,1]) = \{\mathbf{x} = [\underline{x}, \overline{x}] | (\underline{x}, \overline{x}) \in [0,1]^2 \quad \text{y} \quad \underline{x} \leq \overline{x}\}$$

Un conjunto intervalo-valorado difuso Z en el universo $U \neq \emptyset$ es una función $Z : U \rightarrow L([0,1])$.

Una de las interpretaciones existentes de los conjuntos intervalo-valorado difusos, la cual utilizamos en este trabajo, es la siguiente: "el grado de pertenencia de un elemento al conjunto es un valor dentro del intervalo de pertenencia considerado. No conocemos exactamente el valor, por lo que proporcionamos sus extremos" [2].

Siguiendo esta idea, podemos asumir que la amplitud del intervalo representa la ignorancia que tenemos a la hora de asignar el valor de pertenencia del elemento al conjunto.

Aplicándolo sobre nuestro problema de agrupamiento, queremos que si el algoritmo está completamente seguro de que un dato pertenece a un grupo, entonces la amplitud de su intervalo de pertenencia será mínima. No importa si la pertenencia es $[1,1]$ a un grupo y $[0,0]$ a los demás, o si pertenece $[0{,}5, 0{,}5]$ a dos grupos. Por el contrario, si el algoritmo no está seguro de que un dato pertenezca a los grupos que se han creado en los datos, entonces la amplitud de sus intervalos de pertenencia deberá ser mayor. En el caso extremo, si un dato parece no pertenecer a ninguno de los grupos existentes, los intervalos de pertenencia a todos los grupos pueden ser $[0,1]$.

Por tanto, a diferencia del algoritmo Fuzzy Cluster Means, en nuestra propuesta la suma de los extremos inferiores y superiores de las pertenencias de un dato a todos los grupos tiene que ser un valor entre 2 y c, siendo c el número de grupos. En el caso en el que no existe ninguna duda sobre la pertenencia de los datos a los grupos, los extremos inferiores de todos los valores de pertenencia son iguales a los extremos superiores. Manteniendo la misma restricción que existía en el FCM, estos deben sumar 1, por lo que la suma total es 2. En el caso de que la duda sobre la pertenencia sea máxima, los intervalos de pertenencia a todos los grupos serán $[0,1]$, por lo que la suma total será igual al número de grupos.

De forma análoga al k-means y al FCM, con este nuevo algoritmo queremos minimizar la suma ponderada de distancias entre cada dato y los centroides de cada grupo, utilizando como pesos los valores de pertenencia. En este caso, esos valores de pertenencia son intervalos. Cuando no existe duda sobre el valor de pertenencia, el extremo inferior y superior del intervalo son muy parecidos y representan el valor que debe tomar el peso. Por el contrario, cuando existe una gran duda de que un dato pertenezca a un grupo, no queremos que su información modifique en gran medida el valor del centroide, por lo que queremos que su peso sea pequeño. Al ser la amplitud del intervalo grande, eso significa que su extremo inferior tiene que ser pequeño. Por tanto, en ambos casos podemos utilizar un peso para la suma ponderada proporcional al extremo inferior del intervalo de pertenencia.

También es necesario restringir la suma total de las amplitudes de los intervalos de pertenencia. De no hacerlo, nuestro sistema se minimizaría al decir que tenemos una duda máxima sobre la pertenencia de todos los datos existentes.

Por tanto, la función objetivo que queremos minimizar en nuestra propuesta es la siguiente:

$$J = \frac{1}{a}\sum_{k=1}^{n}\sum_{i=1}^{c}(\underline{u_{ik}})^m ||x_k - v_i||_A^2 + \sum_{k=1}^{n}\sum_{i=1}^{c}(\overline{u_{ik}} - \underline{u_{ik}})^m$$

donde x_k es el k-ésimo dato a clasificar, v_i es el centroide del grupo i, $[\underline{u_{ik}}, \overline{u_{ik}}]$ es el intervalo de pertenencia del dato k al grupo i y m es un valor real mayor que 1.

El parámetro $1/a$ permite ajustar la importancia relativa de los dos términos de la ecuación. Hay que tener en cuenta que ambos términos no tienen por que estar en la misma escala: el primer término depende de las distancias entre los datos y el segundo es siempre un valor entre 0 y 1. Mediante este parámetro podemos conseguir que la solución obtenida se parezca más a la obtenida por el FCM si hacemos que el segundo término tenga mucha importancia, o que la solución presente en general amplitudes muy grandes, si es el primer término el más importante.

Esta función está sujeta a las siguientes restricciones:

- Los intervalos tienen que estar bien formados. $\overline{u_{ik}} \geq \underline{u_{ik}}$, $k = 1..n$, $i = 1..c$
- Todos los grupos tienen que tener por lo menos un dato con extremo inferior de pertenencia positivo. $\sum_{k=1}^{n} \underline{u_{ik}} > 0, \quad i = 1..c$
- La suma de los extremos de las pertenencias de un dato a todos los grupos tiene que estar entre 2 y c. $2 \geq \sum_{i=1}^{c}(\underline{u_{ik}} + \overline{u_{ik}}) <= c, \quad k = 1..n$

Cuando el número de grupos es 2, esta función se puede minimizar utilizando multiplicadores de Lagrange. De esta forma, obtenemos un algoritmo iterativo análogo al FCM. A partir de una inicialización aleatoria, podemos actualizar los intervalos de pertenencia basándonos en los datos de los centroides.

$$\underline{u_{ik}} = \frac{2(2a)^{1/m-1}}{||x_k - v_i||^{2/m-1}\left[c + 2(2a)^{1/m-1}\sum_{j=1}^{c}\frac{1}{||x_k - v_j||^{2/m-1}}\right]}$$

$$\overline{u_{ik}} = \frac{2\left[||x_k - v_i||^{2/m-1} + (2a)^{1/m-1}\right]}{||x_k - v_i||^{2/m-1}\left[c + 2(2a)^{1/m-1}\sum_{j=1}^{c}\frac{1}{||x_k - v_j||^{2/m-1}}\right]}$$

para $k = 1..n$, $i = 1..c$. A partir de los datos de los intervalos de pertenencia, podemos actualizar los centroides.

$$v_i = \frac{\sum_{k=1}^{n}(u_{ik})^m x_k}{\sum_{k=1}^{n}(u_{ik})^m}$$

para $i = 1..c$. El proceso termina cuando los cambios en los valores son suficientemente pequeños.

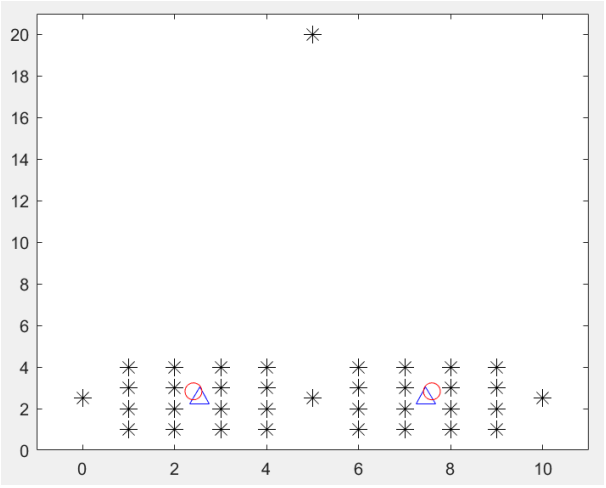

Figura 2. Ejecución del algoritmo propuesto y del FCM sobre un conjunto de datos con un outlier. Conjunto de datos a clasificar marcados con asteriscos negros. Centroides obtenidos por nuestro algoritmo marcados con triángulos azules. Centroides obtenidos por el FCM marcados con círculos rojos.

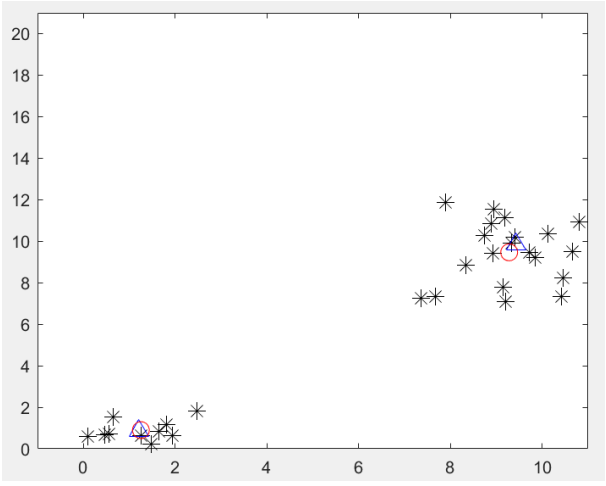

Figura 3. Ejecución del algoritmo propuesto y del FCM sobre un conjunto de datos sin outliers. Conjunto de datos a clasificar marcados con asteriscos negros. Centroides obtenidos por nuestro algoritmo marcados con triángulos azules. Centroides obtenidos por el FCM marcados con círculos rojos.

IV. Ejemplos numéricos

En esta sección mostramos el comportamiento del algoritmo que hemos propuesto. Para poder visualizar los ejemplos con mayor simplicidad y que sean más fáciles de entender, en todos ellos utilizamos sólo dos dimensiones.

Comenzamos con el mismo ejemplo que hemos visto en la Figura 1. Como ya hemos comentado, el FCM no es capaz de resolver este ejemplo correctamente. Al aplicar nuestro algoritmo, los centroides obtenidos ya no se desplazan hacia arriba de donde deberían estar y se colocan en el centro real de los grupos existentes. En la Figura 2 hemos marcado con triángulos azules los centros obtenidos por nuestro algoritmo y con círculos rojos aquellos obtenidos por el FCM.

Si analizamos los intervalos de pertenencia resultantes de este ejemplo, el dato situado en la intersección de los dos grupos (punto (5, 2.5)) tiene pertenencia [0.3488, 0.6512] a cada uno de los grupos, con una amplitud de 0.3024. Por su parte, el dato outlier (punto 5, 20)) tiene pertenencia [0.0004, 0.9996] a cada uno de los grupos, con una amplitud de 0.9991. Podemos ver claramente como el algoritmo es capaz de identificar que estos dos datos no son iguales a la hora de hacer la clasificación. Al tener una amplitud tan grande el outlier, esto hace que casi no se tenga en cuenta a la hora de calcular los centroides, y por eso estos se sitúan en el centro geométrico del grupo correspondiente.

En la Figura 3 podemos ver un nuevo conjunto de datos y cómo se comportan sobre él nuestro algoritmo (centroides marcados con triángulos azules) y el FCM (centroides marcados con círculos rojos). Se puede comprobar que ambos algoritmos obtienen una solución muy similar.

Si a ese mismo conjunto de datos le añadimos tres outliers, entonces vemos que los dos algoritmos ya cambian su comportamiento. Esta nueva ejecución se puede ver en la Figura 4, siguiendo la misma leyenda.

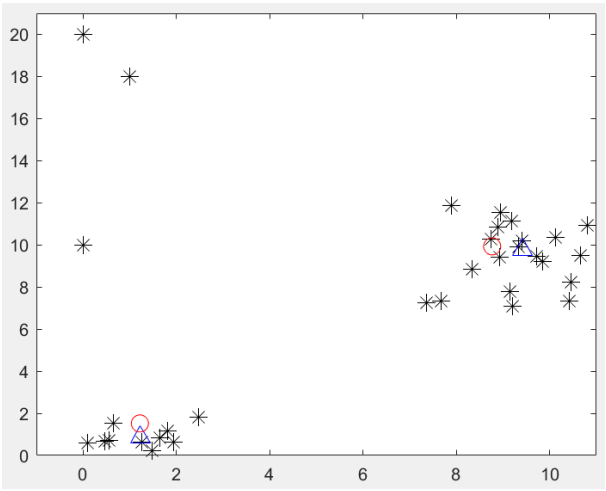

Figura 4. Conjunto de datos igual al de la figura 3 al que se le han añadido tres outliers en los puntos (0,20), (1,18) y (0,10). Ejecución del algoritmo propuesto y del FCM. Conjunto de datos a clasificar marcados con asteriscos negros. Centroides obtenidos por nuestro algoritmo marcados con triángulos azules. Centroides obtenidos por el FCM marcados con círculos rojos.

Como podemos comprobar visualmente, la adición de tres outliers casi no ha modificado los valores de los centroides obtenidos por nuestro algoritmo. Analíticamente, estos valores han pasado de ser

- Grupo1→ (1.2107, 0.8496)
- Grupo2→ (9.4268, 9.8496)

a valer

- Grupo1→ (1.2223, 0.8670)
- Grupo2→ (9.4149, 9.7710)

Sin embargo, en la ejecución del FCM los nuevos centroides se ven claramente influenciados por los outliers, y se desplazan hacia allí. El centroide del grupo 1 se desplaza hacia arriba y el centroide del grupo 2 lo hace hacia la izquierda. De esta forma, pasan de valer

- Grupo1→ (1.2531, 0.8985)
- Grupo2→ (9.2796, 9.4647)

a valer

- Grupo1→ (1.2181, 1.5217)
- Grupo2→ (8.7661, 9.9312)

Uno de los aspectos más importantes a la hora de aplicar nuestro algoritmo de agrupamiento es el ajuste del parámetro $1/a$. Este parámetro permite ajustar la amplitud de los intervalos de pertenencia. Si el parámetro $1/a$ toma valores muy grandes, entonces el algoritmo obtiene soluciones donde las amplitudes de todas las pertenencias son grandes. Por el contrario, si el parámetro $1/a$ toma valores pequeños, entonces el algoritmo encuentra soluciones donde las amplitudes de los intervalos de pertenencia son pequeñas. Para cada uno de los problemas es necesario ajustar este parámetro, para obtener las soluciones deseadas.

En la Figura 5 vemos un nuevo conjunto de datos marcado con asteriscos negros y tres pares de puntos que representan los centros obtenidos con tres valores del parámetro $1/a$ distintos. Mediante cuadrados rojos los centros obtenidos cuando $1/a = 1/0{,}0001$, triángulos azules cuando $1/a = 1/10000$ y círculos verdes cuando $1/a = 1/1$.

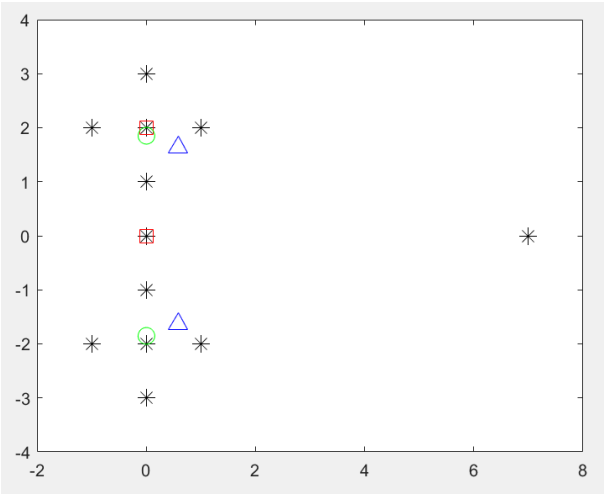

Figura 5. Conjunto de datos marcado mediante asteriscos negros. Centros obtenidos mediante el algoritmo propuesto cuando $1/a = 1/0{,}0001$ (cuadrados rojos), $1/a = 1/10000$ (triángulos azules) y $1/a = 1/1$ (círculos verdes).

Como se puede observar, los resultados varían en gran medida dependiendo del valor del parámetro. En el Cuadro I se muestran los valores de los intervalos de pertenencia de cada dato a cada grupo. Cuando $1/a = 1/0{,}0001$, el algoritmo encuentra dos centroides aleatorios, cuyas pertenencias son $[0, 0]$ y $[1, 1]$ a los dos grupos. El resto de datos tienen una pertenencia de $[0, 1]$ a los dos grupos, es decir, la máxima amplitud posible. En este ejemplo de ejecución, los datos elegidos para convertirse en centroides son el $(0, 2)$ y el $(0, 0)$. Por el contrario, cuando $1/a = 1/10000$, todos los intervalos de pertenencia tienen amplitud 0 y, por ello, pueden ser considerados como valores puntuales. En este caso, el resultado obtenido es muy parecido al obtenido por el FCM. Por último, el caso en el que $1/a = 1/1$ muestra un ejemplo de buen balance entre los dos extremos. En este caso, los datos que están muy cercanos a algún centroide y que, por lo tanto, el algoritmo está seguro de su pertenencia a los grupos existentes, tienen intervalos de pertenencia con amplitudes pequeñas. El dato situado en el punto $(0, 0)$, justo en la intersección de los dos grupos, tiene una amplitud mayor que el resto de datos, puesto que está más alejado de los centroides, pero mucho menor que el dato $(7, 0)$, que no debería pertenecer a ninguno de los dos grupos. Este dato tiene una pertenencia de $[0,0015, 0{,}9985]$ a ambos grupos, casi la máxima posible.

Por tanto, es muy importante ajustar el parámetro $1/a$, que balancea la importancia relativa de las distancias entre los datos y los centroides con las amplitudes de las pertenencias. Según nuestras pruebas hechas sobre varios conjuntos de datos, un valor que funciona en la mayoría de los casos es cuando a toma el valor del percentil 10 o 15 de las distancias entre los datos del conjunto de datos.

V. Conclusiones

En este trabajo hemos presentado una extensión del algoritmo Fuzzy Cluster Means. Mediante nuestra propuesta, utilizamos una extensión de los conjuntos difusos, los conjuntos intervalo-valorados difusos, para permitir que el algoritmo detecte los datos considerados outliers y que estos no interfieran en el proceso de agrupamiento del resto de datos. Mediante varios ejemplos ilustrativos hemos comprobado que el algoritmo se comporta de manera correcta, tanto cuando existen outliers como cuando no. Además, hemos comprobado que es muy importante la elección de un buen valor para el parámetro que pondera los dos términos de la función objetivo, ya que este parámetro permite una variedad total entre soluciones con la máxima y la mínima amplitud en las pertenencias de todos los datos a todos los grupos.

Referencias

[1] J.C. Bezdek, “Pattern Recognition with Fuzzy Objective Function Algorithms” Plenum Press, 1981.
[2] H. Bustince, E. Barrenechea, M. Pagola, J. Fernandez, Z. Xu, B. Bedregal, J. Montero, H. Hagras, F. Herrera, B. De Baets, “A historical account of types of fuzzy sets and their relationships” IEEE Transactions on Fuzzy Systems 24, 179-194, 2016.
[3] E.W. Forgy, “Cluster analysis of multivariate data: efficiency versus interpretability of classifications”, Biometrics 21, 768–769, 1965.
[4] A.K. Jain, M.N. Murty, P.J. Flynn, “Data clustering: A review” ACM Computing Surveys 31, 264-323, 1999.
[5] S.C. Johnson, “Hierarchical clustering schemes” Psychometrika 32, 241-254, 1967.
[6] L. Faufman, P.J. Rousseeuw, “CLustering by means of Medoids”, in Statistical Data Analysis Based on the L_1–Norm and Related Methods, edited by Y. Dodge, North-Holland, 405–416, 1987.
[7] S.P. Lloyd, “Least square quantization in PCM”, IEEE Transactions on Information Theory 28, 129-137, 1982.
[8] J.B. MacQueen, “Some Methods for Classification and Analysis of Multivariate Observations”, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 1, 281-297, 1967.
[9] J.H. Ward Jr., “Hierarchical Grouping to Optimize an Objective Function”, Journal of the American Statistical Association, 58, 236–244, 1963.
[10] L.A. Zadeh, “Fuzzy sets”, Information and Control 8, 338–353, 1965.

Cuadro I

Resultados de los intervalos de pertenencia obtenidos por el algoritmo propuesto sobre el conjunto de datos de la Figura 5. En cada fila se muestra la información referente a un dato. Para él se indican los intervalos de pertenencia a los dos grupos existentes, cuando ejecutamos el algoritmo con diferentes valores del parámetro $1/a$: $1/0{,}0001$, $1/10000$ y $1/1$.

Dato	$1/a = 1/0{,}0001$		$1/a = 1/10000$		$1/a = 1/1$	
	Grupo 1	**Grupo 2**	**Grupo 1**	**Grupo 2**	**Grupo 1**	**Grupo 2**
(-1, -2)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.9725, 0.9725]	[0.0275, 0.0275]	[0.7882, 0.9967]	[0.0033, 0.2118]
(0, -1)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.9897, 0.9897]	[0.0103, 0.0103]	[0.8775, 0.9930]	[0.0070, 0.1225]
(0, -2)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.9988, 0.9988]	[0.0012, 0.0012]	[0.9999, 1.0000]	[0.0000, 0.0001]
(0, -3)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.9899, 0.9899]	[0.0101, 0.0101]	[0.6956, 0.9978]	[0.0022, 0.3044]
(1, -2)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.9995, 0.9995]	[0.0005, 0.0005]	[0.7925, 0.9967]	[0.0033, 0.2075]
(-1, 2)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.0275, 0.0275]	[0.9725, 0.9725]	[0.0033, 0.2118]	[0.7882, 0.9967]
(0, 1)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.0103, 0.0103]	[0.9897, 0.9897]	[0.0070, 0.1225]	[0.8775, 0.9930]
(0, 2)	[1.0000, 1.0000]	[0.0000, 0.0000]	[0.0012, 0.0012]	[0.9988, 0.9988]	[0.0000, 0.0001]	[0.9999, 1.0000]
(0, 3)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.0101, 0.0101]	[0.9899, 0.9899]	[0.0022, 0.3044]	[0.6956, 0.9978]
(1, 2)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.0005, 0.0005]	[0.9995, 0.9995]	[0.0033, 0.2075]	[0.7925, 0.9967]
(0, 0)	[0.0000, 0.0000]	[1.0000, 1.0000]	[0.5000, 0.5000]	[0.5000, 0.5000]	[0.2024, 0.7976]	[0.2024, 0.7976]
(7, 0)	[0.0000, 1.0000]	[0.0000, 1.0000]	[0.5000, 0.5000]	[0.5000, 0.5000]	[0.0015, 0.9985]	[0.0015, 0.9985]

Assessing the performance of bipolar classifiers in three-class problems

1st Guillermo Villarino
Facultad de Estudios Estadísticos
Universidad Complutense de Madrid
Madrid, Spain
gvillari@ucm.es

2nd Daniel Gómez
Facultad de Estudios Estadísticos
Universidad Complutense de Madrid
Madrid, Spain
dagomez@estad.ucm.es

3rd J. Tinguaro Rodríguez
Facultad de Ciencias Matemáticas
Universidad Complutense de Madrid
Madrid, Spain
jtrodrig@mat.ucm.es

Abstract

In the context of supervised classification, several aspects already exist which need to be improved regarding the decision making step that any classifier passes through. Before providing the final assignment, many classification algorithms produce a soft score (either a probability, a fuzzy degree, a possibility, a cost, etc.) assessing the strength of the association between each item to be classified and each class. Usually, the final decision or classification step of these algorithms consists on assigning the item to the class with the highest soft score, a method typically known as the *maximum rule*. However, this procedure does not always take advantage of all the information contained in such soft scores. In other words, the final classification step of many algorithms may be improved through alternative procedures more sensible to the available soft information that the mentioned maximum rule.

To this aim, in this paper we propose a general bipolar approach that enables learning how to take advantage of the soft information provided by many classification algorithms in order to enhance the generalization power and accuracy of the classifiers. To show the suitability of the proposed approach, we also present some computational experiences for three-class classification problems, in which its application to some well-known classifiers as random forest and neural networks produce some improvements in performance.

***Index Terms*—Supervised classification models, bipolar models, Machine learning, Soft information**

I. Introduction

One of the most important topics in data science is classification, and particularly supervised classification tasks. In the literature, there exist a huge diversity of supervised classification algorithms, approaches and applications, depending on the specific tasks, type of data, characteristics or efficiency [7], [8]. Typically, in a supervised classification context the main aim is to be able to classify a set of items into classes based on a training sample or dataset that provides examples of association between items and classes, and that is used to train the classifiers in order to adequately generalize the observed associations, that is, to fit the classification models to the observed data.

Following the ideas presented in [12]–[15], in [17] classical supervised algorithms as CART [2], Random Forest (RF) [3] and Neural Networks [11], [16] were modelled as probabilistic classifiers, providing soft probabilistic assessments of the association of items with classes. In a second step, a bipolar probabilistic representation framework was developed by allowing some opposition or dissimilarity relationships between the classes to be introduced. In a third step, the more convenient structure of dissimilarity relationships was learned through an evolutionary algorithm. This more expressive representational model and the associated learning process permitted to improve the classification performance of the original classifiers in a binary classification context. In this paper we extend these results by addressing three-class classification problems instead of binary ones.

Moreover, in [18] we proposed a replication + aggregation scheme to obtain a fuzzy classifier from a probabilistic one as a robustness enhancing pre-process that permits developing a fuzzy bipolar model from any soft classification algorithm. The experimental results were also carried out in a binary classification context.

The remainder of the paper is organized as follows: Section II describes the preliminary concepts we will use along the work, including the differences between crisp and probabilistic classifiers, as well as some specific concepts regarding accuracy measures and Genetic Algorithms (GAs). Then, in Section III, we present the main idea of bipolar knowledge representation and the complete two-stage (learning and aggregating) process for constructing a bipolar classifier from a soft supervised one. Finally, the experimental framework along with the respective analysis of the results are presented in Sections IV and V. We summarize the paper with the main concluding remarks in Section VI.

II. Preliminaries

In this preliminary section, we introduce some concepts for a better understanding of the paper. We firstly introduce the main concepts of crisp and probabilistic classifiers as well as their differences and relationships to motivate one of the principal contribution of this paper: the importance of modelling the soft information of a classifier before making the final decision in a classification task.

A. Crisp and probabilistic classifiers

Let us denote by $\{C_1, \ldots, C_k\}$ the set classes of a classification problem, and by $X = \{x_1, \ldots, x_n\}$ the set of items that

has to be classified.

As we have pointed in the introduction, many classification users only takes into account the final output of the classification task. This is probably because they are only interested in the final solution provided by the classifier. This is the reason why in a general way, the classifiers are usually viewed as functions

$$C : X \longrightarrow \{C_1, \ldots, C_k\}, \tag{1}$$

that is, a procedure to assign one of the available classes to each of the items being classified.

Nevertheless, the classification process goes through many steps before to arrive to the final assignment, and it is in the intermediate steps that soft information usually appear as a natural way to model the information and the evidence being obtained. Particularly, it is very common that classification algorithms manage soft information for each item $x \in X$ about the probability that x belongs to each of the different classes, or in fuzzy classification models about the degree of membership of the item x in the set of classes.

Taking into account these considerations, in [17] we distinguished between crisp (classical) and probabilistic classifiers. A probabilistic classifier can be viewed as a function

$$C_P : X \longrightarrow [0,1]^k, \tag{2}$$

that assigns to each item x its probability of belonging to each of the available classes. Obviously, for any $x \in X$ it has to hold that $\sum_{i=1}^{k} (C_P(x))_i = 1$ because of the additivity of probability. We would like to remark that many classification algorithms (as for example neural networks, random forest or decision trees) could be viewed as probabilistic classifiers if we just look at the soft information provided by the algorithms before making the final decision or crisp assignment.

III. Probabilistic bipolar model

This section is devoted to present the underlying ideas of bipolar knowledge representation. Firstly, it merits to be stressed that the concept of dissimilarity assumes that the available classes are related through a certain opposition or dissimilarity structure informing of which classes provide negative evidence against the others. This dissimilar structure can be modelled through a dissimilarity matrix D, which contains the degree of dissimilarity for any pair of classes. Obviously, the main diagonal has to be composed by zero values.

It is clear that the dissimilarity matrix D plays a crucial role in this classification scheme since it determines how the negative evidence is derived from the initial evidence for each class. As a consequence, the performance of the resulting crisp classifier, as well as the effect of incorporating the bipolar representation framework and the aggregation method, are absolutely dependent on the choice of the matrix D.

Figure 3 shows a flow diagram of the proposed decision making stage, including the genetic search of the dissimilarity structure and its application to the test set.

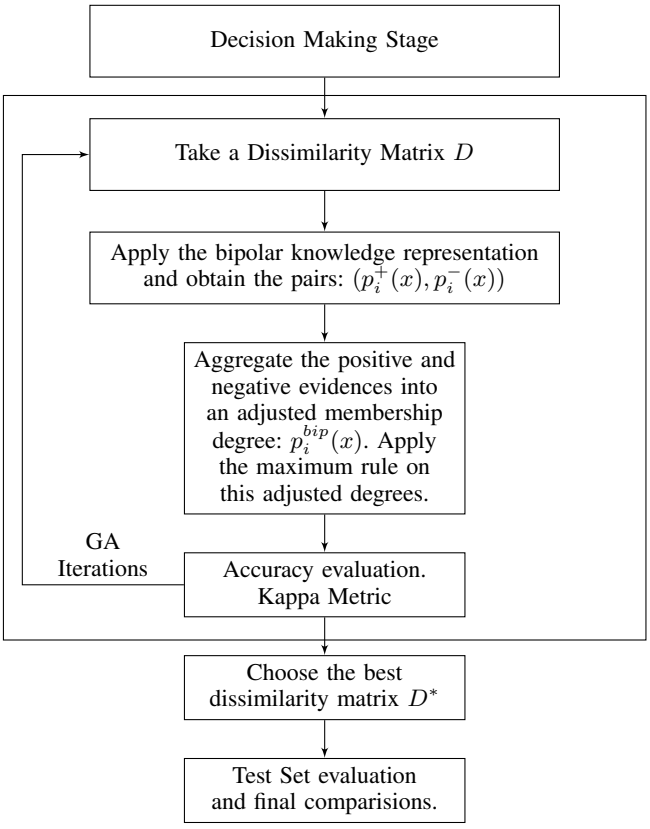

Fig. 1. Flow diagram of the proposed Decision Making Stage.

A. Learning the dissimilarity matrix

Ideally, in real situations the adequate structure of dissimilarity between classes should be proposed by application domain experts based on his knowledge. However, in many cases it may be more practical to obtain the matrix D through a learning process carried out once the base soft classifier has been trained. When this learning process is driven by a measure of performance focused on the generalization accuracy of the adjusted crisp classifier, the resulting matrix tends to fix some of the misassignments committed by the base classifier on the training sample, hopefully also improving its predictive accuracy on new queries or a test sample. Therefore, this learning approach allows that any probabilistic classifier may benefit from introducing a dissimilarity structure in the set of classes, aiding the decision rule of the classifiers to better adapt to the specific features of each dataset or application context.

Here we propose that the learning process of the dissimilarity matrix D is performed by means of a genetic algorithm (GA). The specific parameters of the applied GA are given in Section IV-C.

B. Obtaining the paired structure (p^+, p^-)

In this section we show the application of the dissimilarity matrix already learned by the GA to obtain the paired structure containing the positive and negative evidences.

To do so, we depart from the soft information (estimated probabilities) given by the base algorithm for an item x, $p_i(x) = p_i^+(x)$, treating it as our positive probability of class C_i membership. Then, we apply the bipolar knowledge representation approach to get the negative evidence in the following way:

$$p_i^-(x) = \sum_{j \neq i} d_{ij}\ p_j^+(x) = \sum_{j=1}^{k} d_{ij}\ p_j^+(x) = D_i p^+(x), \quad (3)$$

Once the bipolar paired structure has been obtained, one of the possibilities we have is to aggregate this positive and negative evidences into a bipolar adjusted degree of evidence by applying any kind of aggregation operator.

Let us stress this is only one among the wide spectrum of possibilities for dealing with paired structures.

C. Aggregating bipolar evidence: the additive and logistic cases

Let us now address the question of how to aggregate, for a given class C_i and an item x, the pair of positive and negative evidence degrees $p_i^+(x)$ and $p_i^-(x)$ in order to obtain a single adjusted degree $p_i^{adj}(x)$. Obviously, different aggregation choices will lead to different adjusted classifiers. In this work we have studied two different kinds of aggregation, that are defined below.

Let $p_i^+(x)$, $p_i^-(x)$ be the positive and negative probabilities of item x into class C_i. The additive adjusted degree of x into class C_i is defined as

$$p_i^{add}(x) = \max\{0, p_i^+(x) - p_i^-(x)\}. \quad (4)$$

Notice that the previous definition can be interpreted as the Lukasiewicz t-norm $W(a,b) = \max\{a + b - 1, 0\}$ of the positive and non-negative degrees, that is, $p_i^{add}(x) = W(p_i^+(x), n(p_i^-(x)))$, where n stands for the standard negation $n(a) = 1 - a$. In this way, the positive evidence $p_i^+(x)$ initially provided by the soft classifier is adjusted by subtracting from it the negative evidence $p_i^-(x)$. Particularly, the initial degrees are not modified when no class is dissimilar to C_i, that is, when $D_i = 0$.

Thus, an adjusted degree $p_i^{add}(x) > 0$ represents the existence of a positive gap between the support for class C_i and the support for class dC_i, that is, for the classes considered dissimilar to C_i. In this situation, the strength of the association of item x with class C_i may have been reduced from its initial assessment, but it is still perfectly possible that item x is finally assigned to C_i. On the other hand, a zero value of $p_i^{add}(x)$ represents a situation in which there exist more evidence for the dissimilar class dC_i than for C_i, and thus the adjusted classifier should not assign the item to class C_i.

In the following definition, we propose an alternative way to aggregate the positive and negative information into a single adjusted degree.

Let $p_i^+(x)$, $p_i^-(x)$ be the positive and negative evidence degrees of item x into class C_i. The logistic adjusted membership degree of x into class C_i is defined as

$$p_i^{log}(x) = \begin{cases} 1 - e^{-\frac{p_i^+(x)}{p_i^-(x)}} & if\ p_i^-(x) > 0 \\ 1 & otherwise \end{cases} \quad (5)$$

Unlike the additive logic of the previous aggregation, this logistic aggregation focuses on the ratio between positive and negative information, adjusting it to range in the [0,1] interval through a logistic transformation. This permits a somehow more flexible behaviour of the adjusted degrees, in the sense that the choice of the dissimilarity matrix D may have an even greater influence in the adjustment of the initial positive evidence provided by the base soft classifier, up to the point that class $p_i^{log}(x) = 1$ whenever no evidence is gathered for the dissimilar class dC_i, that is, when $p_i^-(x) = 0$.

As mentioned above, once one of these two aggregation methods has been applied and the adjusted degrees $p_i^{adj}(x)$ has been obtained for each class (either $p_i^{adj}(x) = p_i^{add}(x)$ or $p_i^{adj}(x) = p_i^{log}(x)$), the final decision on the classification of item x is made by applying the maximum rule to such adjusted degrees. Therefore, the item x is finally assigned to the class C_h with a maximum adjusted degree $p_h^{adj}(x)$, that is, $h = arg\max_{i \in \{1,\ldots,k\}} p_i^{adj}(x)$.

IV. Experimental framework

This section is devoted to present a computational experience aimed to assess the performance of our dissimilarity - based bipolar knowledge representation approaches (additive and logistic) when applied on recognized classifiers such as Random Forest [3] and Neural Networks [11], [16].

A. Experimental setting details

As just mentioned, the base classifiers used in this experiment are Random Forest (RF) and Neural Networks (NNet). This experience is designed to compare the benchmark performance of these classifiers with those obtained from the later ones by means of the proposed dissimilarity learning process and the additive and logistic adjustments.

The results for each classifier in each experiment will be obtained following a 5-fold cross validation scheme for each dataset. In each folder, that is, for each training set, the optimal base classifier parametric configuration is approximated using a grid P on the space of parameters of the algorithms considered. In order to evaluate the performance of each specific parametric configuration $p \in P$, 25 bootstrap samples of the training set are generated, in such a way that the base classifiers are fit to each of these bootstrap samples and then tested on a hold-out sample (composed by the non selected instances in the bootstrapping process) using the kappa statistic.

At each folder, the genetic dissimilarity learning process is carried out departing from the vectors of estimated probabilities $p(x)$ of the items x in the training sample in the way shown in III.

The train and test performance measures of each of the 3 classifiers in each dataset considered in each experiment are

Id.	Data-set	#Ex.	#Atts.	(R/I/N)
Aut	Autos	159	25	(15/0/10)
Car	Car	159	25	(15/0/10)
Wnq	Winequality-red	1599	11	(11/0/0)
Pen	Penbased	10992	16	(0/16/0)
Pag	Page-blocks	5472	10	(4/6/0)
Der	Dermathology	366	34	(0/34/0)
Eco	ecoli	336	7	(7/0/0)
Fla	flare	1066	25	(15/0/10)
Gla	Glass	214	9	(9/0/0)
Shu	Shuttle	2175	9	(0/9/10)
Yea	Yeast	1484	8	(8/0/0)
Lin	Lymphography	148	18	(3/0/15)
Bal	Balance	625	4	(4/0/0)
Win	Wine	178	13	(13/0/0)
Nty	Newthyroid	215	5	(4/1/0)
Hay	Hayes-Roth	160	4	(0/4/0)
Con	Contraceptive	1473	9	(6/0/3)
Thy	Thyroid	720	21	(6/0/15)

TABLE I
SUMMARY DESCRIPTION FOR THE EMPLOYED DATASETS.

finally computed by respectively averaging the train and test accuracy rates of the $F = 5$ different folders.

B. Data sets

We have selected a benchmark of 18 datasets from the KEEL dataset repository [1]. Particularly, we have used the 5-folder cross-validation datasets provided by KEEL in the different experiments. Table I summarizes the properties of the selected datasets, showing for each dataset the number of examples (#Ex.), the number of attributes (#Atts.) and type (Real/Integer/Natural) To transform multi-class datasets into trhee-class ones, we have taken as class C_0 and C_1 the originals closest to 20% of instances and as class C_2 the union of the remainder classes.

C. Genetic algorithm details

Finally, regarding the GA used at the evolutionary tuning of the dissimilarity structures, we have used the default GA for real-coded chromosomes implemented in the *genalg* R package. It is a standard GA, with usual crossover and mutation operators, the details of which can be consulted at [20]. The GA has been run with the following configuration, that provided satisfying solutions in a feasible amount of time:

- Population Size: 50 individuals.
- Number of iterations: 20
- Mutation Chance: 0.01.
- Elitism: About 20% of the population size.

Let us note at this point that we have tried a more complex configuration for the GA used in number of iterations, specifically we have used a 40 iterations and 100 individuals with no improvements.

D. Statistical test for performance comparison

In this paper, we use some hypothesis validation techniques in order to give statistical support to the analysis of the results.

Specifically, we employ the Wilcoxon rank test [19] as a non-parametric statistical procedure for making pairwise comparisons between two algorithms. For multiple comparisons, we use the Friedman aligned ranks test, which is recommended in the literature [4], [5] to detect statistical differences among a group of results. Finally, the Holm post-hoc test [6] has been used to find the algorithms that reject the equality hypothesis with respect to a selected control method. A complete description of these tests, with many considerations and recommendations and even the software used to run this analysis can be found on the website http://sci2s.ugr.es/sicidm/.

V. Experimental results

This section is aimed to present the results of the computational experience described above, and carried out in order to study the capacity of enhancement of our bipolar adjusted classifiers with respect to the reference base classifier to which the proposed final decision tuning method is applied.

Results are grouped, for each base algorithm, in pairs for training and test, where the best global result for each considered dataset is stressed in **bold-face**. None is stressed in case of ties.

The experimental study has been obtained using R Software. Specifically, we used the *caret* package [21] for the classifiers training, fitting them through the underlying classical packages *random forest* and *nnet*, and finally the *genalg* package [20] to assess the GA.

For performing all the analysis presented in this paper we have used a computer AMD A10-6700 3.94GHz, 8GB RAM, Windows 8.1.

We can observe from the results of tables II and III the general good behaviour of the bipolar tuning method, at least regarding one of the bipolar adjustment methods, since it allows the improvement in performance of the reference algorithms.

	RF					
	Ref		bipAdd		bipLog	
	Train	Test	Train	Test	Train	Test
Aut	1.000	0.716	1.000	**0.719**	1.000	0.706
Car	0.996	**0.867**	1.000	0.854	1.000	0.857
Wnq	1.000	0.515	1.000	0.489	1.000	**0.525**
Pen	1.000	**0.903**	1.000	0.895	1.000	0.892
Pag	1.000	0.831	1.000	**0.832**	1.000	**0.832**
Der	1.000	**0.995**	1.000	0.993	1.000	0.992
Eco	1.000	0.758	1.000	**0.775**	1.000	0.764
Fla	0.796	0.783	0.805	**0.787**	0.807	0.784
Gla	1.000	0.672	1.000	0.658	1.000	**0.677**
Shu	1.000	**0.996**	1.000	**0.996**	1.000	0.995
Yea	1.000	0.377	1.000	0.366	1.000	**0.378**
Lin	0.981	0.672	0.996	0.675	0.996	**0.710**
Bal	0.612	**0.556**	0.615	0.523	0.617	0.513
Win	1.000	**0.979**	1.000	0.954	1.000	0.973
Nty	1.000	**0.935**	1.000	0.912	1.000	0.895
Hay	0.885	0.703	0.886	**0.715**	0.886	**0.715**
Con	0.788	0.280	0.807	**0.286**	0.807	0.279
Thy	1.000	0.895	1.000	**0.897**	1.000	0.891
Mean	0.948	**0.746**	0.950	0.740	0.951	0.743

TABLE II
RESULTS IN TRAIN AND TEST ACHIEVED BY THE GENETIC BIPOLAR APPROACHES APPLIED TO THE RF ALGORITHM.

Regarding the bipolar method applied to the RF classifier, in Table II we show the results and the following brief description of its behaviour.

- There is no improvement by kappa means when comparing the additive bipolar model against reference.
- The additive bipolar classifier outperforms the classification of the remainder approaches in 8 out of 18 datasets and the logistic one does so in 6 of them.
- Reference wins in 6 out of 18 datasets.
- There is a tie between the additive bipolar approach and the reference in the Shuttle dataset.

Thus we can see that we have reached improvements or ties in 12 out of 18 datasets when comparing. It is important to note the variable behaviour of the additive bipolar method in this case. Despite being the winner method in number of datasets, we can see that its mean is not the best because of the lower kappa value obtained in several of the remainder datasets.

	Nnet					
	Ref		bipAdd		bipLog	
	Train	Test	Train	Test	Train	Test
Aut	0.504	0.382	0.533	**0.385**	0.532	**0.385**
Car	1.000	0.997	1.000	0.997	1.000	0.997
Wnq	0.359	0.341	0.399	**0.356**	0.399	**0.356**
Pen	0.954	0.855	0.964	**0.866**	0.966	0.856
Pag	0.853	0.753	0.874	0.755	0.887	**0.774**
Der	1.000	0.987	1.000	**0.991**	1.000	**0.991**
Eco	0.753	**0.697**	0.779	0.680	0.777	0.688
Fla	0.785	**0.788**	0.794	0.782	0.795	0.777
Gla	0.660	0.507	0.688	**0.517**	0.687	0.513
Shu	0.991	0.976	0.993	**0.977**	0.994	**0.977**
Yea	0.440	0.360	0.473	0.379	0.473	**0.381**
Lin	0.896	0.667	0.922	0.671	0.925	**0.678**
BAl	0.600	0.586	0.603	0.562	0.603	**0.563**
Win	0.945	0.911	0.959	**0.915**	0.960	0.901
Nty	0.986	0.957	0.995	0.957	0.997	0.957
Hay	0.811	**0.615**	0.850	0.600	0.845	0.588
Con	0.356	0.334	0.383	0.336	0.383	**0.338**
Thy	0.859	0.738	0.904	0.770	0.925	**0.803**
Mean	0.764	0.692	0.784	0.694	0.786	**0.696**

TABLE III
RESULTS IN TRAIN AND TEST ACHIEVED BY THE GENETIC BIPOLAR APPROACHES APPLIED TO THE NNET ALGORITHM.

Considering the NNet classifier, the bipolar method reaches the results shown in Table III that could be interpreted as follows:

- There is an improvement by kappa means of 0.004 when comparing the logistic bipolar model against reference, being of 0.002 in case of the additive one.
- Both additive and logistic bipolar classifiers outperform the classification of the remainder approaches in 7 and 10 out of 18 datasets respectively.
- Reference wins in 3 out of 18 datasets.
- There two ties in these results.

On balance we have reached improvements or ties in 14 out of 18 datasets when comparing the bipolar approaches against the reference.

In order to detect significant differences among the results of the different approaches, we carry out the Friedman aligned rank test. This test obtains a low p-value for all the three algorithms, which implies that there are significant differences between the results provided by each method.

For this reason, we can apply a post-hoc test to compare our methodology against the remaining approaches. Specifically, a Holm test is applied using the best approach (the one with lower ranking) as control method and computing the adjusted p-value (APV) for the one with the highest ranking.

Obviously, it would be desirable for the reference to reach the highest or, at least, not the lowest ranking since it is usually associated with worse results.

Algorithm	Rank RF	Rank NNet
"Ref"	22.22	31.5
"BipAdd"	31.83	26.44
"BipLog"	28.44	24.55
p-val	0.00097	0.000913
APV	**0.1336**	**0.371**

TABLE IV
AVERAGE RANKINGS OF THE ALGORITHMS (ALIGNED FRIEDMAN), ASSOCIATED P-VALUES AND HOLM TEST APV FOR EACH ALGORITHM WITH THE MAX AGGREGATION.

Table IV, reflects that there are statistical significant differences between the three classifiers for both RF and NNet algorithms. However, in case of RF this differences and the respective statistical analysis should be carefully interpreted because of the lower ranking value obtained by the reference algorithm. In fact, the reference (RF without applying any bipolar approach) seems to reach the best results regarding the Friedman aligned rank test in spite of not being the best in number of datasets outperformed. Therefore there is no statistical evidence of the superiority of any method compared in case of RF.

Regarding the base Nnet classifier, Table IV shows the superiority of both bipolar approaches in ranking values, however the Holm post-hoc test reflects that there is not enough evidence to ensure that both bipolar approaches outperform the reference.

Comparison	R^+	R^-	p-val
RFbipAdd vs. RFRef	115.0	56.0	0.1913
RFbipLog vs. RFRef	100.0	71.0	0.5135
NNetbipAdd vs. NNetRef	100.0	53.0	0.2559
NNetbipLog vs. NNetRef	95.0	58.0	0.3684

TABLE V
WILCOXON TEST TO COMPARE THE BIPOLAR TUNING APPROACHES (R^+) AGAINST THE BASE CLASSIFIER (R^-).

The statistical analysis of the pairwise comparisons of methods, which is carried out by means of a Wilcoxon test, Table V, reflects the weak superiority of the proposed methodology when it is applied to the RF and Nnet algorithms with not so high p-values in case of additive bipolar model. Again, the application of the methodology on the RF and NNet algorithm does not reach significant improvements.

VI. Discussion and final remarks

In this paper we have studied the extension of probabilistic supervised classifiers into a bipolar knowledge representation framework by means of the introduction of a dissimilarity

structure in the set of classes. These structures enable considering different opposition or dissimilarity relationships between the available classes, that otherwise are by default considered as independent, unrelated objects. These relationships provide further information of the underlying structure of the classification problems being addressed, which can be used at the final decision or classification stage to better exploit the soft scores provided by any classifier to assess the association between each item and each class. Therefore, the introduction of dissimilarity structures may allow to strengthen the adaptation of the classifiers to each specific application context, in which classes acquire a particular semantics, thus also improving the classifier performance.

In this sense, the proposed approach can be understood as a general post processing method to fine tune the maximum decision rule usually applied to make the decision on the class assignment of each item to be classified.

To study the feasibility of the proposed approach, and particularly to remark that it can be applied to any soft classifier despite how it is obtained, we have applied it to two of the most powerful supervised classifiers, random forests and artificial neural networks. A rigorous and extensive computational experience has been conducted to analyse whether the proposed additive and logistic bipolar approaches enabled an statistically significant improvement of the base probabilistic classifiers.

Along this experimental study, we have reached several lessons learned:

- The bipolar framework improved the results of the two base machine learning algorithms considered in this work in number of datasets outperformed.
- Both the additive and logistic adjustment methods did not significantly outperform the results of the base classifier. However, they reached not so high p-values in the Wilcoxon test, specially the additive one.
- Comparing both the additive and the logistic proposed classifiers, we found there is no clear winner. In fact, this question seems to be somehow dependent on the base algorithm considered as well as on the dataset of application.

These results lead us to conclude that the proposed approach provides a suitable solution to confront three-class classification problems and improve the decision rule that manages how the intermediate soft information gathered by many classifiers is exploited.

However, we must improve the results in statistical terms so that we could ensure the superiority of our proposed methodology when applied in three-class classification problems by enlarging the benchmark of datasets, and considering several different parametric configuration for training the base classifier as well as the evolutionary search of the dissimilarity structure among the set of classes.

Regarding future research on this approach, a main line of work will be devoted to study further mechanisms than the additive and logistic aggregations for exploiting the bipolar pairs of positive and negative evidence. A particularly appealing possibility is to use these bipolar pairs as the base information of a multivalued para-consistent logic, as those proposed in [9], [10], [12]. This would allow an even more expressive representational framework to take advantage of all the information contained in the soft scores provided by classifiers.

Acknowledgement

This research has been partially supported by the Government of Spain, grant TIN2015-66471-P and the FPU fellowship grant 2015/06202 from the Ministry of Education of Spain.

References

[1] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, (2011) 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.

[2] Breiman L. (1984) Classification and Regression Trees. New York, NY: Kluwer Academic Publishers;

[3] Breiman L. (2001) Random Forests. Mach. Learn.vol.40 5-32.

[4] Demsar J., (2006) Statistical comparisons of classifiers over multiple datasets, J. Mach. Learn. Res. 7 1-30.

[5] García S., Fernandez A., Luengo J. and Herrera F., (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, Inform. Sciences 180(10) 2044-2064.

[6] Holm S., (1979) A simple sequentially rejective multiple test procedure, Scand. J. Stat. 6 65-70.

[7] Kumar, R. and Verma, R. (2012) Classification algorithms for data mining: A survey, International Journal of Innovations in Engineering and Technology, 2 7-14.

[8] Lim, TS., Loh, WY. and Shih, YS. (2000) A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms, Machine Learning, 40 203-228.

[9] Ozturk,M., Tsoukiàs, A. (2007) Modeling uncertain positive and negative reasons in decision aiding. Decis. Support Syst. 43, 1512-1526

[10] Turunen, E., Ozturk, M., Tsoukiàs, A. (2010) Paraconsistent semantics for Pavelka style fuzzy sentential logic. Fuzzy Sets Syst. 161, 1926-1940

[11] Ripley, B. D. (1996) Pattern Recognition and Neural Networks. Cambridge.

[12] Rodríguez, J.T., Turunen, E.,Ruan,D.,Montero, J. (2014) Another paraconsistent algebraic semantics for Lukasiewicz-Pavelka logic. Fuzzy Sets Syst. 242, 132-147

[13] Rodríguez, J. T., Vitoriano, B., Montero, J. (2011) Rule-based classification by means of bipolar criteria. 2011 IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making (MDCM), 197-204.

[14] Rodríguez, J. T., Vitoriano, B., Montero, J. (2012) A general methodology for data-based rule building and its application to natural disaster management. Computers & Operations Research, 39 (4) 863-873.

[15] Rodríguez JT, Vitoriano B, Gómez D, Montero, J. (2013) Classification of Disasters and Emergencies under Bipolar Knowledge Representation. In: Vitoriano B, Montero J and Ruan D (eds), Decision Aid Models for Disaster Management and Emergencies, vol. 7, Atlantis Computational Intelligence Systems, 209-232.

[16] Venables, W. N. and Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth edition. Springer.

[17] Villarino, G., Gómez, D., Rodríguez, J. T. (2017). Improving Supervised Classification Algorithms by a Bipolar Knowledge Representation. In Advances in Fuzzy Logic and Technology 2017 (pp. 518-529).

[18] Villarino, G., Gómez, D., Rodríguez, J.T. et al. Soft Comput (2018). https://doi.org/10.1007/s00500-018-3320-9

[19] Wilcoxon F., (1945) Individual comparisons by ranking methods, Biometrics 1 80-83.

[20] Willighagen E., (2005) genalg: R Based Genetic Algorithm. http://cran.r-project.org/

[21] Kuhn M., (2008) Building Predictive Models in R Using the caret Package. Journal of Statistical Software, 28(5) 1-26. doi:http://dx.doi.org/10.18637/jss.v028.i05