

Mejorando el clasificador difuso FARC-HD para problemas multi-clase: Estrategia de descomposición OVO utilizando información de todas las clases



Grado en Ingeniería Informática

Trabajo Fin de Grado

Alumno: Imanol Solano Martinez

Director: José Antonio Sanz Delgado

Pamplona, fecha de defensa

Agradecimientos

En primer lugar, quiero agradecer a mi tutor Josean tanto por el interés que ha tenido desde un primer momento en el proyecto, como por esta introducción en el mundo de la investigación y haberme ayudado durante todo este tiempo a que este trabajo salga adelante.

También me gustaría darle las gracias a mi familia y en especial a mis padres, ya que sin su apoyo no habría sido posible llegar hasta aquí.

Por último, me queda agradecer el apoyo y compañía a todos los amigos y amigas que he tenido a lo largo de estos años, tanto en la universidad como fuera de ella, y que han hecho de esta una gran experiencia llena de grandes momentos.

Gracias por todo lo que ha llegado hasta ahora y por lo que está por venir.

Imanol

Resumen

Es sabido que, hoy en día, el campo del *Machine Learning* o Aprendizaje Automático está siendo uno de los más explotados por diferentes científicos de datos en el mundo. Entre los diferentes algoritmos de clasificación existentes en este campo, se encuentra FARC-HD. Un trabajo de investigación anterior consiguió mejorar su rendimiento incluyendo técnicas de descomposición y funciones de solapamiento. En concreto, se utilizó la técnica OVO.

Sin embargo, al incluir dicha técnica, se pierde la interpretabilidad propia de FARC-HD. Por ello, en este trabajo se propone dividir la fase de entrenamiento en dos fases: una primera de extracción de reglas aplicando OVO y una segunda, en la que se aplica a la totalidad de las reglas extraídas en la fase anterior un algoritmo genético cuya función fitness tiene en cuenta el uso de OVO, logrando así una base de datos única e interpretable. Además, se incluye una fase de filtrado que elimina las reglas redundantes para tratar de conseguir también una base de reglas fácilmente interpretable. Con ello, se pretende recuperar la interpretabilidad perdida tratando de mantener la mejora con respecto a FARC-HD.

Palabras clave: Lógica difusa, Sistemas de Clasificación, Reglas de Asociación Difusas, FARC-HD, OVO.

Summary

These days, the field of *Machine Learning* is known to be one of the most popular fields exploited by different experts in data science all around the world. One of the algorithms that can be found in the large number of classification algorithms existing in this field, is FARC-HD. An earlier research work managed to improve its performance by using decomposition techniques and overlap functions, specifically the OVO technique.

However, when including the above-mentioned technique, the innate interpretability of FARC-HD is lost. That is why, in this work, dividing the training phase into two is proposed: one where the rule extraction is done by using the OVO technique, and a second one, where the genetic algorithm whose fitness function considers the use of OVO is applied to all the rules extracted in the previous phase. In this way, a unique and interpretable database is obtained. Also, a filtering phase that removes the redundant rules is included, trying to reach an easily interpretable rule base. The main goal of this proposal, is to recover the lost interpretability from FARC-HD, attempting to maintain the improvement of the previous proposal.

Key words: Fuzzy logics, Classification Systems, Fuzzy Association Rules, FARC-HD, OVO.

Índice de Contenido

Capítulo 1: Introducción y Objetivos	5
1.1. Introducción.....	5
1.2. Objetivos	6
Capítulo 2: Preliminares	7
2.1. Problemas de Clasificación	7
2.2. Lógica Difusa	8
2.3. Funciones Overlap	11
2.4. Algoritmos Genéticos.....	13
2.5. Sistemas de Clasificación Basados en Reglas Difusas	16
2.5.1. Tipos de Reglas	17
2.5.2. Método de Razonamiento Difuso	18
2.6. FARC-HD	19
2.6.1. Proceso de Aprendizaje	20
2.6.2. Proceso de Clasificación	20
2.6.3. Algoritmo Genético en FARC-HD	21
2.7. Técnicas de Descomposición	23
2.7.1. OVO.....	23
2.7.2. OVA.....	26
2.8. FARC-OVO	27
Capítulo 3: FARC-OVO Interpretable	29
3.1. Fase 1: Extracción de Reglas Aplicado Técnicas de Descomposición	30
3.2. Fase 2: Selección Evolutiva	30
3.3. FARC-SEMIOVO	32
3.4. Etapa de Filtrado de la Base de Reglas	32
3.5. Cambio en las Funciones Overlap n-dimensionales	33
Capítulo 4: Marco Experimental.....	34
4.1. Datasets	34
4.2. Configuración de Parámetros y Clasificadores Difusos Utilizados	35
4.3. Medidas de Rendimiento y Pruebas Estadísticas	36

Capítulo 5: Estudio Experimental	37
5.1. Estudiando los modelos FARC-OVO interpretable y FARC-SEMIOVO sin introducir modificaciones en las funciones de solapamiento	38
5.1.1. Análisis de rendimiento de FARC-OVO interpretable y FARC-SEMIOVO	38
5.1.2. Comparativa con FARC-HD y FARC-OVO	40
5.2. Estudiando modelos FARC-OVO interpretable y FARC-SEMIOVO incluyendo modificaciones en las funciones de solapamiento	43
5.2.1. Análisis de rendimiento de FARC-OVO interpretable y FARC-SEMIOVO con las nuevas funciones de solapamiento	43
5.3.2. Comparativa con FARC-HD y FARC-OVO	44
5.3. Efecto de las propuestas en la base de reglas y análisis de la etapa de filtrado .	46
5.3.1. Efecto de FARC-OVO Interpretable y FARC-SEMIOVO en la base de reglas	46
5.3.2. Fase de filtrado	47
Capítulo 6: Conclusiones.....	51
Capítulo 7: Líneas Futuras.....	52
Bibliografía.....	53

Índice de Ilustraciones

<i>Figura 1.</i> Ejemplo lógica clásica.	8
<i>Figura 2.</i> Ejemplo lógica difusa.....	9
<i>Figura 3.</i> Variable difusa junto con sus etiquetas lingüísticas.	9
<i>Figura 4.</i> Esquema algoritmos evolutivos.	13
<i>Figura 5.</i> Esquema de codificación binaria.....	14
<i>Figura 6.</i> Cruzamiento sobre un punto.	14
<i>Figura 7.</i> Cruzamiento uniforme.	15
<i>Figura 8.</i> Método de la ruleta.....	15
<i>Figura 9.</i> Esquema algoritmos genéticos.	16
<i>Figura 10.</i> Diseño de un SCBRD.	17
<i>Figura 11.</i> Particionamiento de la etiquetas lingüísticas en FARC-HD.....	20
<i>Figura 12.</i> Desplazamiento de una etiqueta lingüística.	21
<i>Figura 13.</i> Desplazamiento lateral de la función de pertenencia.	22
<i>Figura 14.</i> Esquema CHC.....	23
<i>Figura 15.</i> Esquema OVO.....	24
<i>Figura 16.</i> Esquema OVA.	27

<i>Figura 17.</i> Esquema FARC-OVO.....	28
<i>Figura 18.</i> Esquema Fase de Extracción de Reglas.....	30
<i>Figura 19.</i> Forma cromosoma CS	31

Índice de Tablas

<i>Tabla 1.</i> Descripción de las características de los dataset utilizados en la experimentación.....	35
<i>Tabla 2.</i> Configuración de parámetros.....	35
<i>Tabla 3.</i> Mejores resultados de los modelos FARC-OVO y FARC-HD con funciones overlap.....	36
<i>Tabla 4.</i> Precisión de los modelos FARC-OVO interpretable y FARC-SEMIOVO si cambios en las funciones overlap.....	39
<i>Tabla 5.</i> Prueba de rangos alineados de Friedman y de Holm para comparar las diferentes funciones de solapamiento sin introducir cambios en ellas.....	40
<i>Tabla 6.</i> Prueba de Wilcoxon para comparar FARC-HD y los mejores modelos de la propuesta.....	41
<i>Tabla 7.</i> Prueba de Wilcoxon para comparar FARC-OVO y los mejores modelos de la propuesta.....	41
<i>Tabla 8.</i> Precisión de los modelos FARC-OVO Interpretable y FARC-SEMIOVO introduciendo las modificaciones en las funciones de solapamiento.	42
<i>Tabla 9.</i> Comparativa media geométrica sin cambio vs con cambio.....	43
<i>Tabla 10.</i> Comparativa media armónica sin cambio vs con cambio.....	44
<i>Tabla 11.</i> Rangos alineados de Friedman y de Holm para comparar las diferentes funciones de solapamiento introduciendo los cambios propuestos.	44
<i>Tabla 12.</i> Prueba de Wilcoxon para comparar FARC-HD con los mejores modelos de la propuesta introduciendo cambios en las funciones de solapamiento.	45
<i>Tabla 13.</i> Prueba de Wilcoxon para comparar FARC-OVO con los mejores modelos de la propuesta introduciendo cambios en las funciones de solapamiento.	45
<i>Tabla 14.</i> Promedio del número de reglas y antecedentes por regla en FARC-HD y FARC-OVO.....	48
<i>Tabla 15.</i> Promedio del número de reglas y de antecedentes por regla en FARC-OVO interpretable y FARC-SEMI-OVO introduciendo los propuestos cambios en las funciones overlap.....	49
<i>Tabla 16.</i> Promedio de número de reglas y número de antecedentes por regla en FARC-SEMIOVO con filtrado.....	50
<i>Tabla 17.</i> Precisión del modelo FARC-SEMIOVO con filtrado.	50

Capítulo 1: Introducción y Objetivos

1.1. Introducción

Dentro del campo de la inteligencia artificial, el *Machine Learning* o aprendizaje automático [1] está siendo uno de los más explotados en la actualidad. Esto se debe a la infinidad de aplicaciones para los que estos algoritmos son capaces de utilizarse con buenos resultados. Entre sus diferentes aplicaciones se encuentran: El filtrado de spam, reconocimiento de dígitos, detección de fraudes, predicciones en bolsa [2], diagnósticos médicos [3], etc.

El aprendizaje automático trata de diseñar algoritmos que aprenden patrones partiendo de una base de datos; y busca que sean capaces de generalizar bien con nuevos datos desconocidos para la máquina.

Existen diferentes tipos de algoritmos de *Machine Learning*. Por un lado, se encuentra el aprendizaje supervisado. En este caso, la salida es conocida para el algoritmo en la fase de entrenamiento. Por otro lado, se encuentran los algoritmos de aprendizaje no supervisado, como los de *clustering*, para los que no se tiene ninguna información acerca de la salida. También existen algoritmos de aprendizaje semi supervisados, por refuerzo, activo, etc.

En este trabajo, se utilizan algoritmos de aprendizaje supervisado. En concreto un algoritmo de clasificación basado en reglas de asociación difusas conocido como FARC-HD (*Fuzzy Association Rule-Based Classification model for High-Dimensional problems*) [4].

Este algoritmo, divide el entrenamiento en tres fases. Una primera, en la que se aplica el algoritmo a priori para generar las reglas. Una segunda, en la que se asocia un valor a cada regla que denota su importancia que servirá preseleccionar las reglas más interesantes. Y, una tercera, en la que se aplica un algoritmo genético para seleccionar el mejor subconjunto de reglas y ajustar las funciones de pertenencia a cada problema de clasificación específico.

Un trabajo posterior [5], utilizó técnicas de descomposición tales como las técnicas OVA (*One vs All*) y OVO (*One vs One*) y funciones de solapamiento [5] para tratar de mejorar el rendimiento de FARC-HD en problemas multi-clase. Al aplicar OVO junto con algunas funciones de solapamiento se conseguía mejorar notablemente el rendimiento de dicho sistema. Sin embargo, esta propuesta tenía dos hándicaps:

- Se pierde la interpretabilidad que ofrecen los sistemas de clasificación basados en reglas al introducir técnicas de descomposición.
- Por las propias características del OVO, los clasificadores base no tienen en cuenta los ejemplos que no incluyen al par de clases que van a clasificar, perdiendo así la información que estos proporcionan para el resto de las clases. Esto puede provocar que, ante nuevos ejemplos, el sistema produzca resultados engañosos. Es decir, que los clasificadores no son competentes [6].

Este hecho, hace plantearse si se puede lograr, por un lado, una mejora del algoritmo original de FARC-HD manteniendo su interpretabilidad al aplicar técnicas de descomposición. Y, por otro lado, tratar de solventar el problema de los clasificadores no competentes del modelo OVO. Logrando así, superar o por lo menos igualar el modelo FARC-OVO pero añadiendo la interpretabilidad perdida.

1.2. Objetivos

El objetivo principal de este proyecto es el de tratar de recuperar la interpretabilidad propia del modelo FARC-HD [4] perdida al introducir las técnicas OVO manteniendo esa mejora en la precisión a la hora de afrontar los problemas de clasificación multi-clase lograda con dicha técnica de descomposición, obteniendo unos resultados similares a los conseguidos en FARC-OVO [5].

Para ello, la solución propuesta consiste en aplicar el algoritmo evolutivo de FARC-HD [4] al conjunto total de reglas extraídas en los clasificadores binarios de OVO. Una característica del método propuesto es que se consigue solventar el problema de los clasificadores no competentes [6] anteriormente mencionado.

Capítulo 2: Preliminares

En esta sección se va a detallar el marco teórico en el que se engloba este TFG, para lo que se describirán los conceptos básicos necesarios para entender la propuesta realizada. Finalmente, se detallará la propuesta previa [5] que ha sido la motivación de este proyecto.

2.1. Problemas de Clasificación

Los problemas de clasificación son uno de los problemas que los algoritmos de *Machine Learning* tratan de resolver. En este tipo de problemas consisten en discernir a qué clase pertenece cada uno de los ejemplos o patrones que el sistema pueda recibir en un futuro. Para lograrlo, se parte de un conjunto de ejemplos para lo que su salida (su clase) es conocida, con intención de poder aprender un clasificador capaz de clasificar correctamente ejemplos que le son desconocidos. Es decir, que sea capaz de generalizar correctamente con patrones o ejemplos de la vida real.

En función del número de clases que componen la salida, se pueden distinguir dos tipos de problemas: binarios (aquellos que constan de dos clases) y multi-clase (con más de dos clases). En el caso de FARC-OVO, y en este TFG, se trabajará con problemas multi-clase. Es aquí donde entran las denominadas estrategias de descomposición, que tratan de dividir el problema multi-clase original en subproblemas binarios más sencillos de resolver, combinando la salida de estos clasificadores binarios denominados clasificadores base para predecir la clase a la que pertenece un ejemplo. En la literatura especializada, diferentes estrategias de descomposición han sido propuestas [7]. Entre las más conocidas se encuentran las denominadas técnicas OVA (*One vs All*) [8, 9] y OVO (*One vs One*) [10]. En concreto, en este trabajo se hace uso de la técnica OVO.

Además, es necesario establecer una medida de rendimiento que permita medir la bondad del clasificador. Existen diferentes tipos de medidas de rendimiento para calcular la precisión o exactitud de un modelo. La métrica más utilizada es el grado de precisión o *accuracy rate*, definida como el porcentaje de ejemplos correctamente clasificados con relación al número total. También existen otras métricas como la *precisión* (número de ejemplos clasificados como clase positiva o clase de interés, que realmente son de la clase positiva) o el *recall* (ejemplos de la clase positiva clasificados correctamente).

A la hora de entrenar estos modelos, es necesaria una metodología de evaluación que permita validar los sistemas de predicción. Para ello, se crean conjuntos de ejemplos de entrenamiento y test. Los primeros, son utilizados para realizar el aprendizaje del modelo, mientras que el conjunto de test sirve para estimar la calidad del modelo ante nuevas situaciones. Es decir, mide la capacidad de generalización del modelo. Existen distintas técnicas para obtener estos conjuntos: *hold-out*, *leave-one-out*, *bootstrapping* y *validación cruzada*. En este caso, como se verá posteriormente, se considera el modelo de *validación cruzada de 5 particiones*. Para ello, se divide el conjunto original de datos en 5 particiones, cada una de ellas conteniendo el 20% de los ejemplos. Se emplea una combinación de 4 de ellos para el entrenamiento (80%) y el resto es utilizado para probar el modelo (conjunto de test). Así, el resultado para un conjunto de datos se obtiene calculando la media aritmética de las cinco particiones.

2.2. Lógica Difusa

La lógica difusa fue introducida en 1965 por Lotfi Zadeh [11]. Al contrario que en la lógica clásica, donde un elemento puede pertenecer o no a un conjunto, es decir, únicamente puede tomar el valor verdadero o falso. En lógica difusa, se emplean valores reales en el rango $[0,1]$ que denotan en que grado pertenece dicho elemento a un conjunto difuso.

Por ejemplo, supóngase un problema en el que se trata de determinar si un paciente tiene o no fiebre, es decir, si su temperatura corporal está en el conjunto de que denotaremos *Fiebre*. En lógica clásica, existiría un límite a partir del cual se determinaría si el paciente tiene o no fiebre, una temperatura a partir de la cual se podría decir que la temperatura de dicho paciente pertenece al conjunto *Fiebre fuerte*. Véase **Figura 1**.

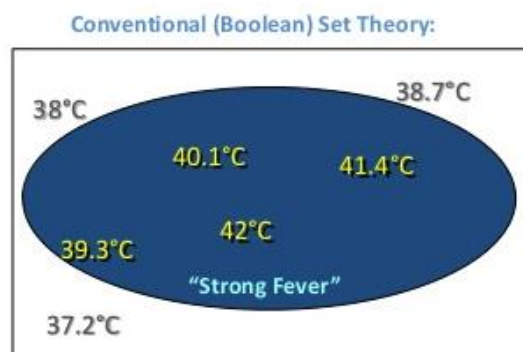


FIGURA 1. EJEMPLO LÓGICA CLÁSICA.

En lógica difusa, en cambio, se elimina dicho concepto del límite, de manera que ya no se diría que el paciente tiene o no fiebre, sino que se asignaría un valor real entre $[0,1]$ llamado *grado de pertenencia* que determina en que grado pertenece dicha temperatura al conjunto *Fiebre fuerte*. Véase **Figura 2**.

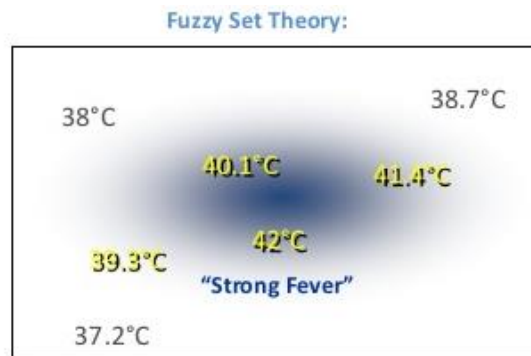


FIGURA 2. EJEMPLO LÓGICA DIFUSA.

Estos conjuntos permiten representar conceptos lingüísticos de manera más cercana a la realidad.

Se denomina a aquellas variables que son utilizadas para representar conjuntos difusos, así como su estado, *variables difusas* y *etiqueta lingüística* a aquellos valores que pueden tomar las variables difusas (**Figura 3**).

Linguistic variables

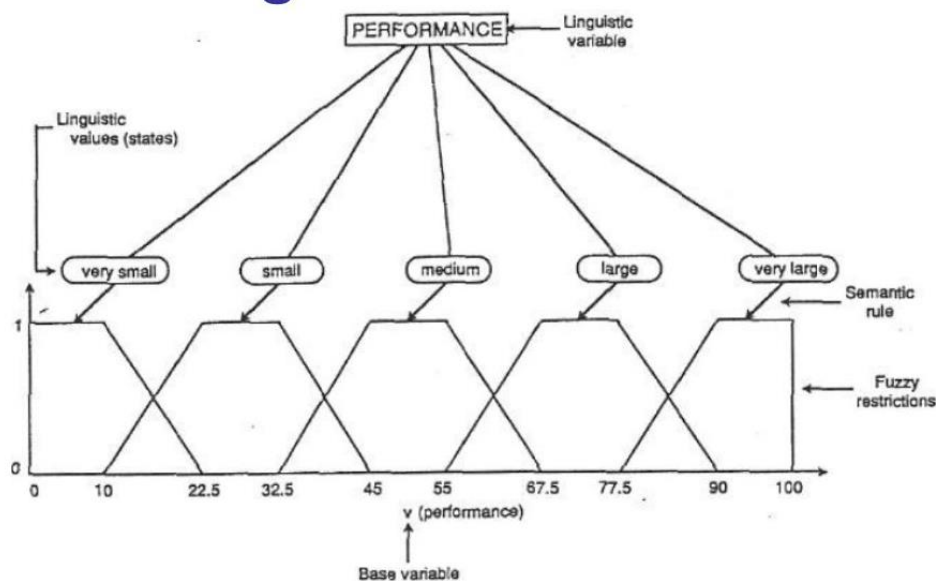


FIGURA 3. VARIABLE DIFUSA JUNTO CON SUS ETIQUETAS LINGÜÍSTICAS.

Al igual que en la lógica clásica donde existen los operadores de unión, intersección y complementario, es necesario definir una serie de métodos para poder realizar dichas operaciones entre conjuntos difusos, conocidas como *t-normas*, *t-conormas* y *negaciones*.

T-normas

Definición 1 Una *t-norma* es una función $T: [0,1]^2 \rightarrow [0,1]$ que satisface:

1. Cota: $T(x, 1) = x \quad \forall x \in [0,1]$
2. Monotonía: $T(x, y) \leq T(z, u)$ si $x \leq z$ e $y \leq u$
3. Conmutativa: $T(x, y) = T(y, x)$
4. Asociativa: $T(T(x, y), z) = T(x, T(y, z))$

Estos operadores son utilizados para representar la intersección. Algunos ejemplos son:

- Mínimo: $T(x, y) = \min \{x, y\}$
- Producto: $T(x, y) = x * y$
- Lukasiewicz: $T(x, y) = \max \{0, x + y - 1\}$

T-conormas

Definición 2 Una *t-conorma* es una función $S: [0,1]^2 \rightarrow [0,1]$ que satisface:

1. Cota: $S(x, 0) = x \quad \forall x \in [0,1]$
2. Monotonía: $S(x, y) \leq S(z, u)$ si $x \leq z$ e $y \leq u$
3. Conmutativa: $S(x, y) = S(y, x)$
4. Asociativa: $S(S(x, y), z) = S(x, S(y, z))$

Estos operadores son utilizados para representar la unión. Algunos ejemplos son:

- Máximo: $S(x, y) = \max \{x, y\}$
- Suma algebraica: $S(x, y) = x + y - x * y$
- Lukasiewicz: $S(x, y) = \min \{1, x + y\}$

Negaciones

Definición 3 $c: [0,1] \rightarrow [0,1]$ es una negación difusa si y solo si:

1. $c(0) = 1$ y $c(1) = 0$
2. $c(x) \leq c(y)$ si $x \geq y$

Además, se dice que una negación es estricta si:

1. $c(x)$ es continua
2. $c(x) < c(y)$ si $x > y$

E involutiva si:

1. $c(c(x)) = x$

Cuando una negación es estricta e involutiva se dice que es una negación fuerte.

Así, dado un conjunto A , su complementario es $A_c = c(\mu_A(u))$ siendo c una negación fuerte.

2.3. Funciones Overlap

Las *funciones overlap* o *funciones de solapamiento* [12] fueron empleadas por primera vez en el procesamiento de imagen con el objetivo de clasificar aquellos píxeles para los que no estaba claro si pertenecía al fondo o a la imagen. Estas, han sido aplicadas en numerosos problemas de procesamiento de imagen [13, 14]. Sin embargo, las funciones overlap han demostrado ser muy útiles en otros ámbitos, ya que permiten recuperar muchas de las características de las t-normas sin imponer la asociatividad.

Definición 4 [12] Una función $O: [0,1] \times [0,1] \rightarrow [0,1]$ se considera una función de solapamiento si satisface las siguientes condiciones:

1. $O(x, y) = O(y, x) \quad \forall x, y \in [0,1]$
2. $O(x, y) = 0$ si y sólo si $x \cdot y = 0$
3. $O(x, y) = 1$ si y sólo si $x \cdot y = 1$
4. Es creciente.
5. Es continua.

En FARC-OVO [5], se definió una extensión a las funciones de solapamiento bidimensionales definidas anteriormente, a n dimensionales:

Definición 5 [5] Una función n -dimensional $O^n: [0,1]^n \rightarrow [0,1]$ con $n \geq 2$ es una función de solapamiento n -dimensional si se mantienen las siguientes propiedades:

1. O^n es simétrica.
2. $O^n(x_1, \dots, x_n) = 0$ si y sólo si $\prod_{i=1}^n x_i = 0$
3. $O^n(x_1, \dots, x_n) = 1$ si y sólo si $\prod_{i=1}^n x_i = 1$
- 4.
5. O^n es creciente.
6. O^n es continua en cada una de las variables.

En FARC-OVO [5], se sustituyeron a las t -normas utilizadas en FARC-HD [4] para calcular el grado de emparejamiento por las siguientes overlap n -dimensionales:

- *Producto (PROD):*

$$O^n(x_1, \dots, x_n) = \prod_{i=1}^n x_i \quad (2.1)$$

- *Mínimo (MIN):*

$$O^n(x_1, \dots, x_n) = \min(x_1, \dots, x_n) \quad (2.2)$$

- *Media armónica (HM):*

$$O^n(x_1, \dots, x_n) = \begin{cases} \frac{n}{\frac{1}{x_1} + \dots + \frac{1}{x_n}} & \text{si } x_i \neq 0, \forall i = 1, \dots, n \\ 0 & \text{en caso contrario} \end{cases} \quad (2.3)$$

- *Media geométrica (GM):*

$$O^n(x_1, \dots, x_n) = \sqrt[n]{\prod_{i=1}^n x_i} \quad (2.4)$$

- *Seno (SIN):*

$$O^n(x_1, \dots, x_n) = \sin\left(\frac{\pi}{2} \left(\prod_{i=1}^n x_i\right)^\alpha\right) \quad \text{donde } \alpha \leq \frac{1}{2n} \quad (2.5)$$

En este TFG, también se hará uso de todas las overlap mencionadas anteriormente, a excepción del seno, ya que daba malos resultados en [5].

2.4. Algoritmos Genéticos

Los Algoritmos Genéticos (AG) [15, 16] son uno de los distintos modelos computacionales de los denominados Algoritmos Evolutivos (AE) propuestos dentro del campo de la Computación Evolutiva (CE) junto con las Estrategias Evolutivas (EE) y la Programación Evolutiva (PE), tal como muestra la **Figura 4**. Todos ellos, tratan de modelar la evolución y tienen las siguientes características en común:

- Utiliza un proceso de aprendizaje colectivo de una población de individuos en la que cada uno de ellos codifica un punto del espacio de búsqueda.
- Utilizan procesos de mutación y cruce para generar descendientes.
- A cada individuo se le asigna una medida de calidad llamada fitness que será utilizada para la selección y competición entre individuos, favoreciendo a aquellos con mejor fitness.

A su vez, la forma en la que se codifican los individuos, así como el diseño de los operadores de mutación y cruce y el mecanismo de selección constituyen la diferencia entre los distintos tipos de AEs.

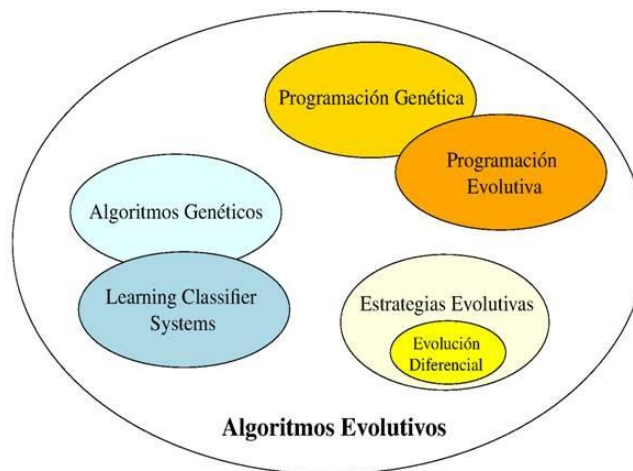


FIGURA 4. ESQUEMA ALGORITMOS EVOLUTIVOS.

Como ya se ha mencionado los AGs [15, 16] son uno de los tipos de AEs existentes. Hay tres características que distinguen a los AGs del resto de AEs tal como se concibieron en un principio:

- El esquema de codificación binario.
- El método de selección: La selección proporcional.
- El método de cruce.

Siendo principalmente el método de cruce lo que los hace diferentes del resto de AEs. Además, en propuestas posteriores se han utilizado métodos alternativos para la codificación y la selección. A continuación, detallan los elementos de *codificación*, *cruce*, *mutación* y *selección* utilizados en los AG, junto con algunos ejemplos:

Codificación

La codificación es la forma en la que se representa la información en nuestro cromosoma, es decir, el modo en que se relaciona el problema en el mundo real con el algoritmo. Uno de los más utilizados es el esquema de *representación binaria*, en el que cada cromosoma se codifica en una secuencia de ceros y unos:

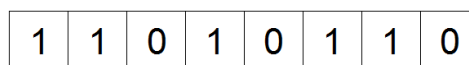


FIGURA 5. ESQUEMA DE CODIFICACIÓN BINARIA.

Existen también representaciones no binarias, tales como la *representación real*, que codifica el cromosoma en una secuencia de reales. La *representación entera*, que lo codifica en una secuencia de enteros, o la *representación mediante alfanuméricos*, mediante una secuencia de caracteres.

Operadores de Cruce

Los operadores de cruce se encargan de generar nuevos individuos para las siguientes generaciones. Es decir, a partir de dos cromosomas padres, genera uno o más descendientes para la próxima generación, permitiendo al AG evolucionar. Entre los diferentes operadores se encuentran el *cruzamiento sobre un punto*: dados dos cromosomas de longitud N, se elige un punto entre 0 y N-1 y se mezclan a partir de dicho punto:

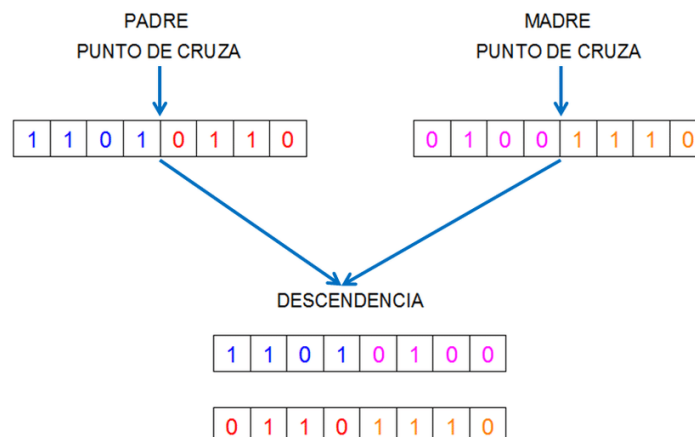


FIGURA 6. CRUZAMIENTO SOBRE UN PUNTO.

O el *cruzamiento uniforme*: dados dos cromosomas, se considera cada gen del descendiente de forma individual, y se decide aleatoriamente de qué progenitor va a heredarlo:

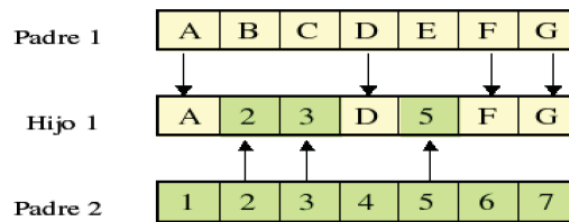


FIGURA 7. CRUZAMIENTO UNIFORME.

Operadores de Mutación

Estos operadores actúan sobre cada individuo y lo modifican ligeramente (intercambiar genes, generar nuevos aleatoriamente, ...). Junto con los operadores de cruzamiento, son de los más importantes, puesto que hacen que en cada iteración la población pueda evolucionar.

Algunos ejemplos son la *mutación por intercambio*, que intercambia los valores de dos posiciones del cromosoma aleatoriamente, o la *mutación por inversión*, que invierte el orden de los genes de una parte del cromosoma.

Métodos de Selección

Los métodos de selección tratan de escoger los individuos mejor adaptados de la población para generar la siguiente población aplicando los *operadores de cruzamiento*. Uno de los más conocidos y empleados es el *método de la ruleta*. En este, se divide el intervalo unidad $[0, 1]$ en tantos subintervalos como progenitores haya. Además, estos el tamaño subconjuntos dependen directamente de la probabilidad que tiene cada progenitor de ser seleccionado. Así, cuanto mejor adaptado esté un progenitor, más probabilidades tiene de ser seleccionado. Para seleccionar un progenitor, basta con generar un número aleatorio entre $[0, 1]$ y ver a qué cromosoma pertenece dicho subintervalo.

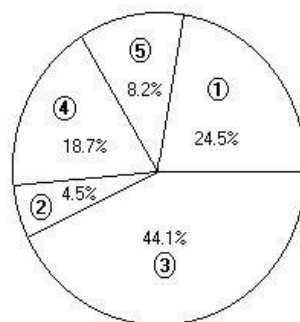


FIGURA 8. MÉTODO DE LA RULETA.

El esquema general de un AG es el siguiente (véase **Figura 9**):

1. Se parte de una población inicial, que codifica mediante alguna representación genética las posibles soluciones iniciales. El número de individuos en la población inicial se elige en un principio y se mantiene siempre igual. Estos individuos se denominan cromosomas.
2. Se evalúa la población actual con una función fitness.
3. Se seleccionan los cromosomas que darán lugar a la nueva generación.
4. Se realizan las operaciones de cruce y mutación de los cromosomas seleccionados.
5. Se genera la nueva población. En esta, se mantiene el mejor o los mejores cromosomas de la anterior población (*elitismo*) y el resto son reemplazados por lo generados en el paso anterior.
6. Este proceso de evaluación, selección, cruce y mutación se repite hasta llegar a una condición de parada.

```

Algoritmo Genetico
inicio
     $t \leftarrow 0$ 
    Inicializar ( $P(t)$ )
    Evaluar ( $P(t), f$ )
    Mientras Parada( $P(t), \Theta_p$ )  $\neq$  Verdadero Hacer
         $t \leftarrow t + 1$ 
        Seleccionar ( $P(t-1), f, \Theta_s, P(t)$ )
        Recombinar( $P(t), \Theta_r, P'(t)$ )
        Mutar ( $P'(t), \Theta_m, P''(t)$ )
         $P(t) \leftarrow P''(t)$ 
        Evaluar ( $P(t), f$ )
    fin_mientras
    Devolver el mejor individuo  $C$  o la población  $P$ 
fin
    
```

FIGURA 9. ESQUEMA ALGORITMOS GENÉTICOS.

2.5. Sistemas de Clasificación Basados en Reglas Difusas

Los Sistemas de Clasificación Basados en Reglas Difusas (SCBRD), son clasificadores utilizados para resolver problemas de clasificación puesto que, además de ofrecer resultados precisos, proporcionan modelos interpretables. Esto se debe al uso de términos lingüísticos en los antecedentes de las reglas, utilizados habitualmente por los usuarios de la aplicación

El esquema de este tipo de clasificador se muestra en la **Figura 10**. En concreto, esta está formada por dos componentes principales:

1. **Base de Conocimiento (BC):** Este componente, almacena toda la información aprendida a partir de una base de datos de un problema específico. A su vez, está compuesta por:
 - **Base de Datos (BC):** Contiene la definición de los conjuntos difusos asociados a los términos lingüísticos utilizados por la Base de Reglas.
 - **Base de Reglas (BR):** Formada por un conjunto de L reglas difusas de clasificación. Estas reglas pueden ser de diferentes tipos, como se verá más adelante.

$$R = \{R_1, \dots, R_L\}$$

2. **Método de Razonamiento Difuso (MRD):** Utiliza la información contenida en la Base de Conocimiento para determinar la clase a la que pertenece cualquier patrón de datos admisible. Para ello, aprovecha la potencia del razonamiento aproximando propia de los sistemas difusos.

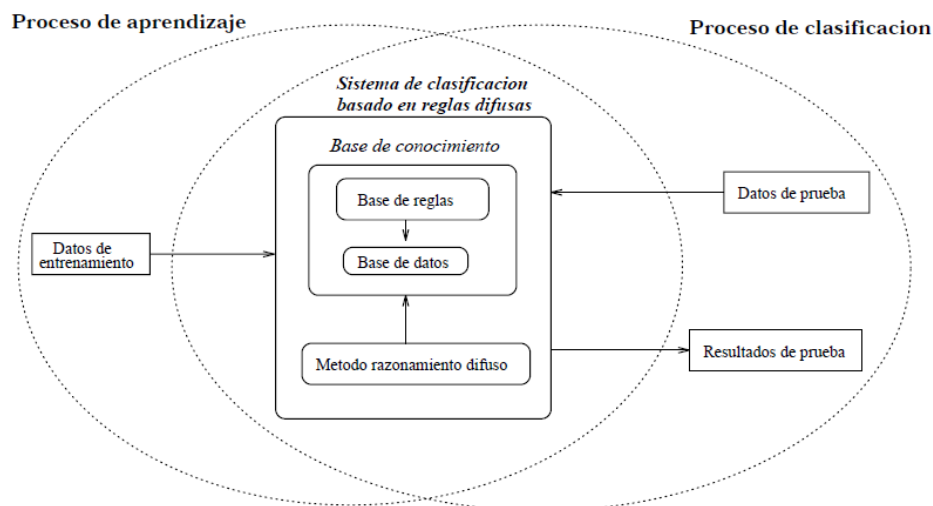


FIGURA 10. DISEÑO DE UN SCBRD.

2.5.1. Tipos de Reglas

El proceso de obtención de la BC debe extraer un conjunto de reglas difusas que representen de la forma más precisa posible el problema que se está tratando. Para ello, existen dos aspectos a determinar: La partición difusa utilizada para cada variable (Base de Datos) y la generación de reglas (Base de Reglas). Los tipos de reglas utilizados en la literatura especializada para SCBRD son los siguientes:

a. Reglas difusas con una clase en el consecuente

$$R_k: \text{Si } X_1 \text{ es } A_{11}^k \text{ y } \dots \text{ y } X_N \text{ es } A_{N1}^k \text{ entonces } Y \text{ es } C_j$$

Donde X_1, \dots, X_N son las variables de entrada, $A_{11}^k, \dots, A_{N1}^k$ son las etiquetas lingüísticas de las variables e Y es la variable de salida que indica a cuál de las clases C_j del conjunto de clases pertenece el ejemplo.

b. Reglas difusas con una clase y un grado de certeza asociado

$$R_k: \text{Si } X_1 \text{ es } A_{11}^k \text{ y } \dots \text{ y } X_N \text{ es } A_{N1}^k \text{ entonces } Y \text{ es } C_j \text{ con grado } r^k$$

Donde X_1, \dots, X_N son las variables de entrada, $A_{11}^k, \dots, A_{N1}^k$ son las etiquetas lingüísticas de las variables, Y es la variable de salida que indica a cuál de las clases C_j del conjunto de clases pertenece el ejemplo y r^k es el grado de certeza de la regla R_k .

c. Reglas difusas con grados de certeza asociados a cada clase del consecuente

$$R_k: \text{Si } X_1 \text{ es } A_{11}^k \text{ y } \dots \text{ y } X_N \text{ es } A_{N1}^k \text{ entonces } (r_{11}^k, \dots, r_{M1}^k)$$

Donde X_1, \dots, X_N son las variables de entrada, $A_{11}^k, \dots, A_{N1}^k$ son las etiquetas lingüísticas de las variables y el consecuente es un vector de pesos en el que cada r_{j1}^k denota el grado de certeza de la regla R_k para predecir la clase C_j .

2.5.2. Método de Razonamiento Difuso

El MRD es un procedimiento de inferencia que utiliza la información de la BC para predecir una clase ante un ejemplo no clasificado. Es decir, indica cómo debe utilizarse la información contenida en la BR, y es por ello, uno de los elementos más importantes en un SCBRD, puesto que determinará su rendimiento.

Sea $x_p = (x_{p1}, \dots, x_{pn})$ un nuevo ejemplo a clasificar. Generalmente, el método que sigue un MRD para clasificar un ejemplo x_p es el siguiente:

1. *Grado de emparejamiento.* Se calcula el grado de emparejamiento del ejemplo x_p con los antecedentes de cada regla de la BR:

$$\mu_{A_j}(x_p) = T\left(\mu_{A_{j1}}(x_{p1}), \dots, \mu_{A_{jn_j}}(x_{pn_j})\right) \quad \text{con } j = \{1, \dots, L\} \quad (2.6)$$

Siendo $\mu_{A_{ji}}(x_{pi})$ el grado de pertenencia del ejemplo con el antecedente i -ésimo de la regla, T es una t -norma [17] y n_j es el número de antecedentes de la regla R_j .

2. *Grado de asociación.* El grado de asociación del patrón x_p con cada regla de la BR se calcula utilizando la siguiente expresión:

$$b_j(x_p) = \mu_{A_j}(x_p) \cdot RW_j \quad (2.7)$$

Donde RW_j es el peso de la regla j -ésima, R_j .

3. *Grado de confianza.* A continuación, se calcula el grado de confianza para cada clase. Se pueden utilizar diferentes funciones de agregación para calcularlo. Las más típicas son la del máximo (MRD de la regla ganadora) y la suma (MRD de combinación aditiva) [18].
4. *Clasificación.* Se predice la clase con grado de confianza más alto.

$$Clase = \arg \max_{l=1, \dots, m} (conf_l(x_p)) \quad (2.8)$$

2.6. FARC-HD

Uno de los SCBRDs más precisos e interpretables de la literatura es el algoritmo FARC-HD (Fuzzy Association Rule-based Classification model for High-Dimensional problems) [4]. El algoritmo emplea la siguiente estructura de reglas:

$$\text{Regla } R_j: \text{ Si } x_1 \text{ es } A_{j1} \text{ y } \dots \text{ y } x_n \text{ es } A_{jn} \text{ entonces Clase} = C_j \text{ con } RW_j \quad (2.9)$$

donde R_j es la etiqueta de la regla j -ésima, $x = (x_1, \dots, x_n)$ representa un ejemplo a través de un vector n -dimensional, A_{ji} representa un conjunto difuso, $C_j \in \mathbb{C}$ es la etiqueta de la clase y RW_j es el peso de la regla.

El peso RW_j , es calculado mediante el valor de confianza difusa o grado de certeza definido en [19]:

$$RW_j = CF_j = \frac{\sum_{x_p \in \text{Class } C_j} \mu_{A_j}(x_p)}{\sum_{p=1}^N \mu_{A_j}(x_p)} \quad (2.10)$$

donde $\mu_{A_j}(x_p)$ hace referencia al grado de emparejamiento del ejemplo x_p con los antecedentes de la regla difusa calculado mediante la Ecuación (2.6). En este caso, se utiliza la t -norma producto.

En el caso de FARC-HD, las etiquetas lingüísticas se modelan empleando funciones de pertenencia triangulares uniformemente distribuidas, las cuales forman una partición fuerte como se muestra en la **Figura 11**. Estas etiquetas son de tipo *do not care*, es decir, que no todas las variables se utilizan en cada regla, por lo que el número de antecedentes de cada regla puede ser diferente.

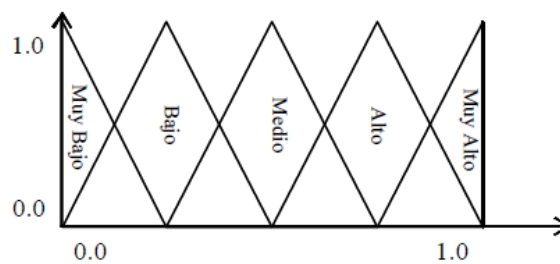


FIGURA 11. PARTICIONAMIENTO DE LA ETIQUETAS LINGÜÍSTICAS EN FARC-HD.

2.6.1. Proceso de Aprendizaje

El proceso de aprendizaje llevado a cabo en FARC-HD está compuesto de 3 etapas:

1. *Extracción de reglas difusas para clasificación*: Se emplea el algoritmo de extracción de reglas *a priori* [20]. A través de un árbol de búsqueda, se obtienen las reglas difusas para cada clase. El número de antecedentes queda limitado por la profundidad máxima del árbol.
2. *Filtrado de reglas candidatas*: Por medio de un esquema de ejemplos ponderados [21], se seleccionan las reglas más interesantes de entre las obtenidas en la fase anterior. Esto, reduce notablemente el coste computacional de la siguiente fase.
3. *Selección evolutiva de reglas*: Se emplea un algoritmo genético (CHC) [22] para ajustar los conjuntos difusos y seleccionar las reglas más precisas partiendo de la base de reglas obtenida en las anteriores fases.

2.6.2. Proceso de Clasificación

FARC-HD aplica un Método de Razonamiento Difuso denominado *combinación aditiva* [18]. El proceso utilizado para clasificar un ejemplo $x_p = (x_{p1}, \dots, x_{pn})$ es el descrito en el apartado **2.5.2. Método de Razonamiento Difuso**.

Como se ha mencionado anteriormente, la t -norma utilizada en el cálculo del *grado de emparejamiento* en el producto. Para calcular el *grado de confianza*, se utiliza el método de la suma:

$$conf_l(x_p) = \sum_{R_j \in RB; C_j=l} b_j(x_p), l = 1, 2, \dots, m \quad (2.11)$$

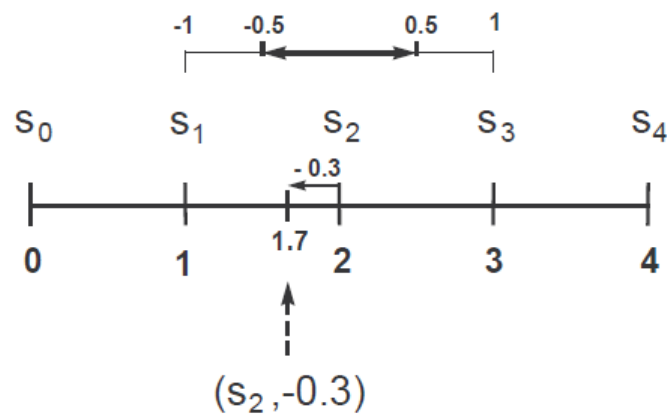
2.6.3. Algoritmo Genético en FARC-HD

A continuación, se describen las características del AG utilizado por FARC-HD en su etapa de selección evolutiva:

- 1) *Codificación de los Cromosomas*: Se utiliza un esquema de doble codificación, de manera que una parte del cromosoma se utiliza para la selección de reglas (C_S) y la otra para el ajuste de las etiquetas (C_T).

Para C_S , se utiliza una codificación binaria, donde '1' indica que la regla ha sido seleccionada y '0' lo contrario. De forma que $C_S = \{c_1, \dots, c_L\}$ siendo L el número de reglas candidatas.

Para C_T en cambio, se codifica mediante una representación real el parámetro α que indica el desplazamiento de las etiquetas lingüísticas (**Figura 12**).



a) Symbolic Translation of a Linguistic Term

FIGURA 12. DESPLAZAMIENTO DE UNA ETIQUETA LINGÜÍSTICA.

En este desplazamiento de las etiquetas lingüísticas, se desplaza la función de pertenencia (el triángulo en este caso) lateralmente, a izquierda o derecha, sin permitir que las etiquetas intercambien sus posiciones (**Figura 13**).

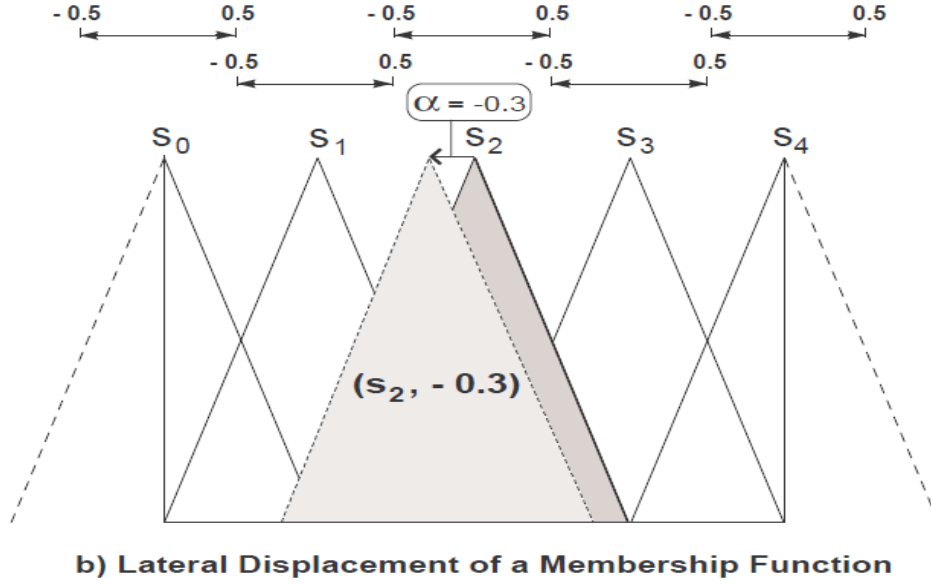


FIGURA 13. DESPLAZAMIENTO LATERAL DE LA FUNCIÓN DE PERTENENCIA.

Teniendo que (m^1, m^2, \dots, m^n) indica el número de etiquetas por cada una de las n variables, se tiene que $C_T = (c_{11}, \dots, c_{1m^1}, c_{21}, \dots, c_{2m^2}, \dots, c_{n1}, \dots, c_{nm^n})$. Así, un cromosoma estaría codificado de la siguiente manera: $C = \{C_S, C_T\}$.

- 2) *Evaluación de los cromosomas*: Para la evaluación de los cromosomas se emplea la siguiente función fitness:

$$Fitness(C) = \frac{\#Hits}{N} - \delta \cdot \frac{NR_{initial}}{NR_{initial} - NR + 1.0} \quad (2.12)$$

Donde $\#Hits$ es el número de ejemplos correctamente clasificados, $NR_{initial}$ es en número de reglas candidatas, NR es el número de reglas seleccionadas y δ es un parámetro que determina la compensación entre el porcentaje de acierto y la complejidad, dado por un $\{C_S, C_T\}$ experto.

- 3) *Operadores de cruce*: El operador a utilizar viene determinando por la parte del cromosoma para la que se aplica.

Para C_T , se utiliza un operador denominado *Parent Centric BLX* (PCBLX) [23]. Este, está basado en el concepto de vecindad, que permite a los genes resultantes del cruce estar en una zona próxima a la de los progenitores.

Para C_S , se utiliza el operador *Half Uniform Crossover Scheme* (HUX) [24], que intercambia la mitad de los genes que son diferentes en ambos progenitores.

- 4) *Modelo Evolutivo CHC*: FARC-HD utiliza el modelo evolutivo CHC [22] (**Figura 14**). Éste, hace uso de mecanismos de selección de la población para una búsqueda global adecuada. También utiliza mecanismos de prevención de incestos, es decir, dos cromosomas solo se cruzan si la distancia de *hamming* está por encima de un umbral determinado (inicializado con el máximo de las distancias entre progenitores), y un proceso de reinicio cuando el umbral, que decrece cuando no se generan nuevos individuos en una iteración, es menor que cero.

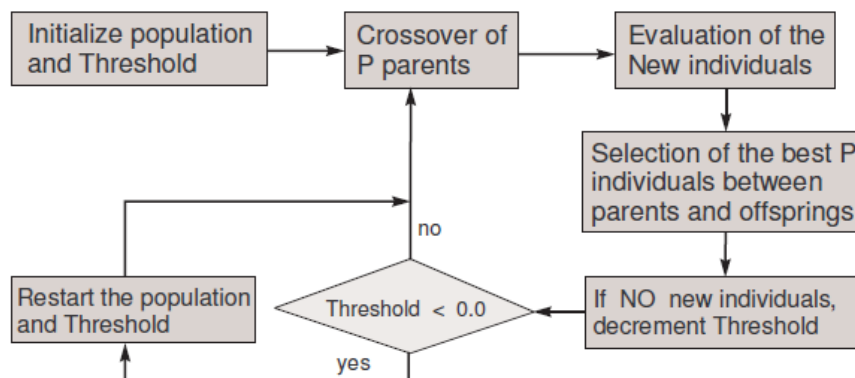


FIGURA 14. ESQUEMA CHC.

2.7. Técnicas de Descomposición

El objetivo de las técnicas de descomposición [7] es el de afrontar problemas multiclase con clasificadores binarios. Para ello, se descompone el problema principal en diferentes subproblemas en los que únicamente hay que diferenciar entre dos clases; haciéndolo más sencillo de resolver, y aprender un clasificador para cada subproblema. A estos se los denomina clasificadores base. Tras ello, para clasificar un nuevo ejemplo, se agregan las salidas de cada clasificador base.

Entre las técnicas de descomposición más utilizadas se encuentran las denominadas OVO (One vs One) [10] y OVA (One vs All) [8, 9].

2.7.1. OVO

La técnica OVO, divide en problema multiclase original en tantos subproblemas como pares de clases haya. De manera que, si el problema original consta de m clases, se tendrán $m*(m-1)/2$ clasificadores base binarios. En la **Figura 15** se muestra el esquema de método para un problema con tres clases.

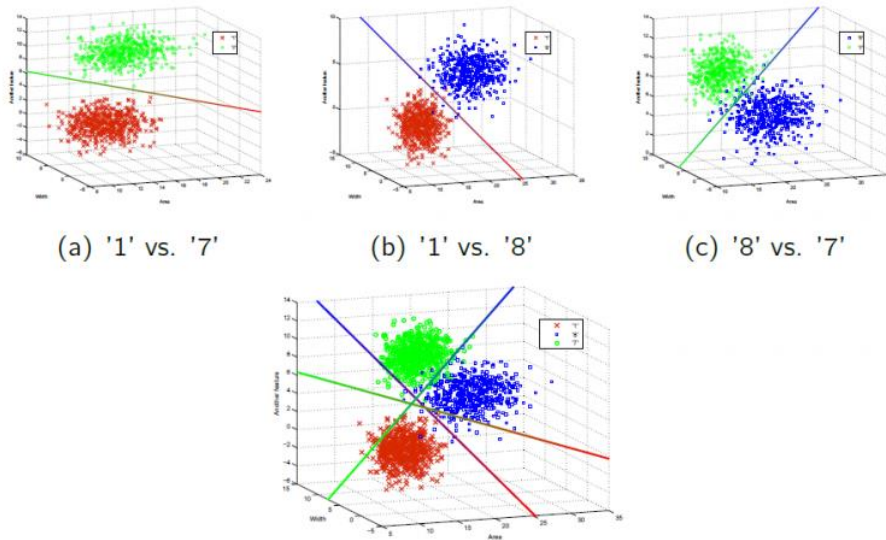


FIGURA 15. ESQUEMA OVO.

En el entrenamiento de cada clasificador base binario, se enfrenta una clase con otra, de manera que cada clasificador se encarga de distinguir el par de clases $\{C_i, C_j\}$.

Para clasificar nuevos ejemplos, los clasificadores devuelven el par de grados de confianza $r_i, r_j \in [0, 1]$ en favor de las clases C_i y C_j respectivamente. Esta información es almacenada en la matriz de votos R .

$$R = \begin{pmatrix} - & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & - \end{pmatrix} \quad (2.13)$$

En casos en los que el clasificador base no es probabilístico (como es FARC-HD), es necesario normalizar dicha matriz para que el rango de las confianzas que proporciona cada clasificador base sea el mismo. La matriz de votos se normaliza como sigue:

$$r_{ij} = \begin{cases} \frac{r_{ij}}{r_{ij} + r_{ji}} & \text{si } r_{ij} \neq 0 \text{ or } r_{ji} \neq 0 \\ 0.5 & \text{si } r_{ij} = r_{ji} = 0 \end{cases} \quad (2.14)$$

Por último, después de que cada clasificador base haya votado, se agregan sus salidas para obtener la clase resultante. En este trabajo se consideran los siguientes métodos de agregación:

- *Voto simple (VOTE)* [25]

$$Class = arg \max_{i=1,\dots,m} \sum_{1 \leq j \neq i \leq m} S_{ij} \quad (2.15)$$

donde S_{ij} es 1 si $r_{ij} > r_{ji}$ y 0 en caso contrario.

- *Voto ponderado (WV)* [26]

$$Class = arg \max_{i=1,\dots,m} \sum_{1 \leq j \neq i \leq m} r_{ij} \quad (2.16)$$

- *Learning Valued Preference for Classification (LVPC)* [27]

$$Class = arg \max_{i=1,\dots,m} \sum_{1 \leq j \neq i \leq m} P_{ij} + \frac{1}{2} C_{ij} + \frac{N_i}{N_i + N_j} I_{ij} \quad (2.17)$$

donde N_i es el número de ejemplos de la clase i en el conjunto de entrenamiento, C_{ij} es el grado de conflicto (grado en que ambas clases están soportadas), I_{ij} es el grado de ignorancia (grado en que ninguna de las clases esté soportada), y P_{ij} y P_{ji} son las preferencias estrictas para i y j , respectivamente. Estas variables se calculan como sigue:

$$\begin{aligned} C_{ij} &= \min\{r_{ij}, r_{ji}\} \\ P_{ij} &= r_{ij} - C_{ij} \\ P_{ji} &= r_{ji} - C_{ij} \\ I_{ij} &= 1 - \max\{r_{ij}, r_{ji}\} \end{aligned}$$

- *WinWV* [5]

$$Class = arg \max_{i=1,\dots,m} \sum_{1 \leq j \neq i \leq m} s_{ij} \quad (2.18)$$

donde s_{ij} es r_{ij} si $r_{ij} > r_{ji}$ y 0 en caso contrario.

Así, este método de descomposición permite dividir el problema original en subproblemas más pequeños (binarios) y más sencillos de resolver. Esto, hace que los subproblemas sean computacionalmente más rápidos y se obtenga una mayor precisión.

Sin embargo, existe un inconveniente inherente a este método de descomposición: los clasificadores no competentes [6]. En el proceso de aprendizaje, el clasificador base únicamente tiene en cuenta los ejemplos correspondientes a una de las dos clases que distingue, ignorando el resto. Esto provoca que dichas clases sean desconocidas para los clasificadores y sus salidas sean irrelevantes al clasificar ejemplos del par de clases que se trata de distinguir. A la hora de clasificar en cambio, estas salidas se agregan de igual manera, lo que puede causar una clasificación incorrecta.

2.7.2. OVA

La técnica OVA, divide el problema multiclase original en tantos subproblemas binarios como clases. De manera que, si el problema original consta de m clases, se tendrán también m clasificadores binarios. Véase **Figura 16**.

Para entrenar cada clasificador base, se toman los ejemplos de una de las clases del problema como la primera de las clases del clasificador base binario, y el resto de los ejemplos del conjunto de entrenamiento, como si fueran una única clase, para representar la segunda clase del clasificador binario.

Tras ello, para clasificar un nuevo ejemplo, cada clasificador base devuelve un grado de confianza $r_i \in [0, 1]$ en favor de cada clase C_i utilizada como primera clase en su entrenamiento. Esta información se guarda en el vector de votos R .

$$R = (r_1, \dots, r_i, \dots, r_m) \quad (2.19)$$

Este vector, en casos en los que el clasificador base no es probabilístico (como es FARC-HD), es necesario normalizarlo al igual que con OVO.

Por último, se utiliza un operador de agregación para obtener la clase resultante. El más utilizado es el del *máximo*.

$$Class = \arg \max_{i=1, \dots, m} r_i \quad (2.20)$$

De esta manera, con el método OVA se consigue solventar el problema de los clasificadores no competentes [6], puesto que se utiliza la totalidad de los ejemplos en cada entrenamiento. Además, la fase de agregación luego del entrenamiento de cada uno de los clasificadores es más simple. Sin embargo, en este caso los problemas que tiene que resolver cada clasificador son más complejos y tienen un alto costo computacional.

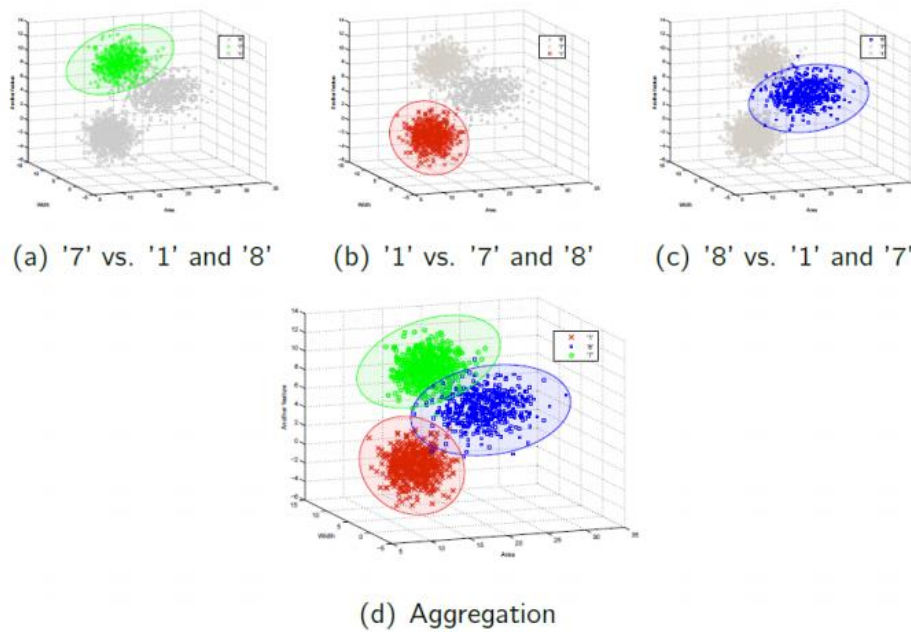


FIGURA 16. ESQUEMA OVA.

2.8. FARC-OVO

Este cambio al algoritmo original de FARC-HD consiste en hacer uso de la estrategia de descomposición OVO, combinándolo con el algoritmo FARC-HD original [5]. Para ello, se rellena la matriz de votos (Ecuación (2.13)) con las confianzas dadas por FARC-HD para cada clase, calculados mediante la ecuación (2.11). Los grados de emparejamiento y asociación se calculan mediante las Ecuaciones (2.6) y (2.7) respectivamente. Tras ello, se realiza la agregación mediante una de las ecuaciones descritas anteriormente (Ecuaciones (2.15), (2.16), (2.17) y (2.18)).

Sin embargo, debido al uso de la t-norma producto, cuando se agregan valores bajos, el rango en el que el resultado puede variar es pequeño. Esta situación se da al calcular el grado de emparejamiento de un ejemplo con varias reglas, así como cuando se tiene un número de antecedentes alto en las reglas. Esto último, provoca que los grados de asociación sean más pequeños y tengan una variación menor. A su vez, implica que las confianzas almacenadas en la matriz de votos tengan una variación más pequeña, un comportamiento poco deseable en el proceso de agregación realizado en OVO.

Así pues, en el proceso de inferencia, se hace uso de funciones de solapamiento n-dimensionales en lugar de las antes mencionadas t-normas. De esta manera, se logra que los resultados obtenidos a la hora de agregar los valores involucrados en el cálculo del grado de asociación se encuentren en un rango más amplio. Manteniendo así más información en el proceso de agregación con OVO. Algunos ejemplos de funciones de solapamiento son:

- *Producto (PROD):*

$$O^n(x_1, \dots, x_n) = \prod_{i=1}^n x_i \quad (2.16)$$

- *Mínimo (MIN):*

$$O^n(x_1, \dots, x_n) = \min(x_1, \dots, x_n) \quad (2.17)$$

- *Media armónica (HM):*

$$O^n(x_1, \dots, x_n) = \begin{cases} \frac{n}{\frac{1}{x_1} + \dots + \frac{1}{x_n}} & \text{si } x_i \neq 0, \forall i = 1, \dots, n \\ 0 & \text{en caso contrario} \end{cases} \quad (2.18)$$

- *Media geométrica (GM):*

$$O^n(x_1, \dots, x_n) = \sqrt[n]{\prod_{i=1}^n x_i} \quad (2.19)$$

- *Seno (SIN):*

$$O^n(x_1, \dots, x_n) = \sin\left(\frac{\pi}{2} \left(\prod_{i=1}^n x_i\right)^\alpha\right) \quad \text{donde } \alpha \leq \frac{1}{2n} \quad (2.20)$$

Por el contrario, esta propuesta, tiene el inconveniente de que se pierde la interpretabilidad intrínseca que ofrece FARC-HD. Esto se debe al hecho de tener más de un clasificador, ya que se tienen tantas Bases de Datos y de Reglas como clasificadores binarios tiene el problema, como puede verse en la **Figura 17**. Esto conlleva a que un mismo término lingüístico puede estar definido de muchas formas diferentes en función de la Base de Reglas en la que se use, ya que el algoritmo genético de cada problema binario los optimizará de formas diferentes.

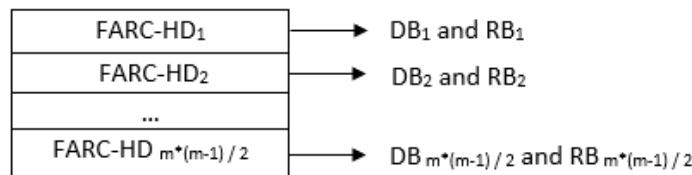


FIGURA 17. ESQUEMA FARC-OVO.

Capítulo 3: FARC-OVO Interpretable

En este TFG, se propone modificar el modelo FARC-OVO ya existente de manera que; por un lado, se recupere la interpretabilidad propia del algoritmo original (FARC-HD). Por otro lado, se tratará de solventar el inconveniente de los clasificadores no competentes propio del método OVO [6], incluyendo la información perteneciente al resto de clases en la etapa de entrenamiento.

Hasta ahora, el proceso de entrenamiento de cada clasificador binario se lleva a cabo aplicando FARC-HD, es decir, a través de las etapas de:

1. *Extracción de reglas difusas para clasificación.*
2. *Filtrado de reglas candidatas.*
3. *Selección evolutiva de reglas y ajuste de la posición de las etiquetas.*

Sin embargo, en esta nueva propuesta se dividirá dicho proceso de manera que, en una primera fase se apliquen las etapas de *extracción y filtrado de reglas* (etapas 1 y 2), y una segunda fase que realiza la *selección evolutiva* (etapa 3). Este nuevo modelo, denominado *FARC-OVO interpretable*, aplica el algoritmo evolutivo en la segunda fase de manera global. Para ello, se fusionan las reglas de cada todos los clasificadores base extraídas anteriormente, pero únicamente utilizando una base de datos, por lo que todas las reglas utilizarán la misma BD, haciéndola única al problema. Esta BD es la que se optimiza con el evolutivo, de forma que en la función de fitness se tiene en cuenta que el proceso de clasificación se hace aplicando OVO.

En este capítulo, se describe el modo en que se divide el proceso de aprendizaje en las dos fases antes mencionadas. En primer lugar, se define la primera fase, en la que se extrae el conjunto de reglas de cada clasificador binario. Esta primera fase es común a todas las propuestas que se realizan (Sección **3.1. Fase 1: Extracción de Reglas Aplicado Técnicas de Descomposición**). Después, se define la segunda fase del entrenamiento, es decir, el cambio realizado en el algoritmo genético, que permite incorporar la información del resto de las clases durante el entrenamiento de los clasificadores binarios (Sección **3.2. Fase 2: Selección Evolutiva**). Además, se introduce una variante al modelo *FARC-OVO interpretable* propuesto. En esta, tras realizar la primera fase, se aplica el algoritmo genético a la totalidad de las reglas extraídas anteriormente, omitiendo la segunda fase del entrenamiento; es decir, en el algoritmo evolutivo no se aplica la votación realizada con OVO, sino que se aplica el MRD original de FARC-HD [4]. Este método se llamará *FARC-SEMIOVO* a partir de ahora (Sección **3.3. FARC-SEMIOVO**). Una vez hecho, se utilizarán ambos modelos durante la experimentación (**Capítulo 4: Marco Experimental**).

3.1. Fase 1: Extracción de Reglas Aplicado Técnicas de Descomposición

Sea $C_{CLASIF} = \{c_1, \dots, c_l\}$ el conjunto de clasificadores binarios generados por OVO, siendo l el número total de clasificadores ($m*(m-1)/2$).

En esta primera fase, por cada clasificador binario $c_i \in C_{CLASIF}$ con $i = 1, \dots, l$ se aplican las dos primeras etapas de FARC-HD de *extracción* y *el filtrado de reglas difusas*.

Así, al final del proceso, se tiene una Base de Reglas (BR) por cada clasificador base. Sea BR_{c_i} la base de reglas extraída para el clasificador base $c_i \in C_{CLASIF}$. Bajo estas condiciones, se denota como $BR_{TOTAL} = \{BR_{c_1}, \dots, BR_{c_l}\}$ al conjunto formado por todas aquellas BR extraídas para cada clasificador base. Donde $BR_{c_i} = \{r_{i1}, \dots, r_{i|BR_{c_i}|}\}$ y r_{ij} hace referencia a la regla j -ésima de la Base de Reglas del clasificador binario i -ésimo. Este conjunto, BR_{TOTAL} será el utilizado posteriormente al aplicar el algoritmo genético.

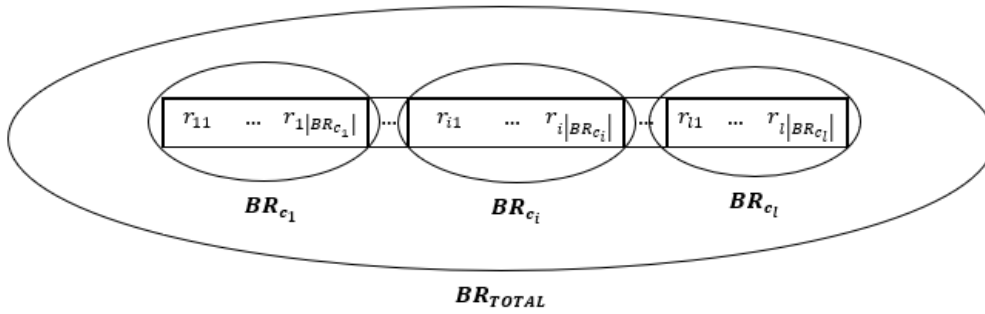


FIGURA 18. ESQUEMA FASE DE EXTRACCIÓN DE REGLAS.

La Base de Datos (BD) es común al conjunto C_{CLASIF} . De esta manera, se tiene una base de datos que es única para todos los clasificadores y se recupera la interpretabilidad perdida al introducir el método OVO en FARC-HD.

3.2. Fase 2: Selección Evolutiva

Como se ha visto anteriormente (**2.6.3. Algoritmo Genético en FARC-HD**), el AG utilizado por FARC-HD es el encargado de realizar tanto la selección de reglas como el ajuste de las etiquetas lingüísticas. En este proyecto, se introducen una serie de cambios en relación con la primera; la selección de reglas, manteniendo la parte de ajuste de etiquetas lingüísticas tal como se define en el algoritmo original.

En concreto, los cambios necesarios que permiten introducir la información del resto de clases en el modelo OVO, tienen que ver con la codificación del cromosoma para la selección de reglas (C_S) y la etapa de evaluación. No afecta, en cambio, a los operadores de selección y cruce, que se mantienen igual.

Con respecto al cromosoma C_S , el cambio es mínimo, puesto que, al tener reglas provenientes de diferentes clasificadores binarios, extraídas en la fase anterior, únicamente afecta al tamaño de este. Además, ha de tenerse en cuenta también, para la posterior etapa de evaluación, a qué clasificador binario $c_i \in C_{CLASIF}$ pertenece cada subconjunto de C_S .

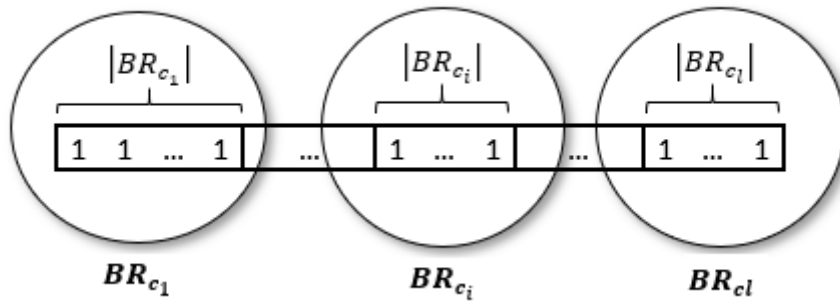


FIGURA 19. FORMA CROMOSOMA C_S .

Sea la fórmula mostrada en la Ecuación 2.15 la función fitness a maximizar. En esta, se tiene $\#Hits$ como parámetro que hace referencia al número de ejemplos correctamente clasificados. A la hora de clasificar un ejemplo, en el modelo actual, se utiliza el cromosoma C_S en su totalidad para hacer la selección de reglas. Con la nueva propuesta, en cambio, esto no es posible, ya que se tiene que la BR inicial (BR_{TOTAL}) está compuesta por reglas provenientes de diferentes clasificadores y, por tanto, se rompería con el modelo OVO propuesto. Para ello, es necesario modificar el método utilizado para clasificar cada ejemplo, de manera que se consiga aplicar la votación realizada con las técnicas OVO pero, en este caso, se usa la información de todas las clases para optimizar el proceso.

Sea $x_p = (x_{p1}, \dots, x_{pn})$ un ejemplo a clasificar, C_S el cromosoma utilizado para la selección de reglas y $BR_{TOTAL} = \{BR_{c_1}, \dots, BR_{c_l}\}$ el conjunto formado por las BR de cada clasificador $c_i \in C_{CLASIF}$. Se propone el siguiente método para clasificar los ejemplos:

1. *Clasificar el ejemplo empleando uno de los clasificadores de BR_{TOTAL} .* Para ello, se utiliza una de las $BR_{c_i} \in BR_{TOTAL}$ junto con su correspondiente subsecuencia del cromosoma C_S , que indica qué reglas de BR_{c_i} son seleccionadas para clasificar el ejemplo. Este proceso de clasificación se lleva a cabo tal como es descrito en FARC-HD (2.6.2. **Proceso de Clasificación**).

2. *Rellenar la matriz de votos con las confianzas logradas en el paso anterior.* La matriz de votos se rellena tal como indica el método OVO (Ecuación 2.2).
3. *Repetir los pasos 1 y 2 hasta haber recorrido cada $BR_{c_i} \in BR_{TOTAL}$.*
4. *Agregar los valores para obtener la salida, y contabilizar el acierto o fallo.* Se utiliza un tipo de voto como los descritos anteriormente (**2.7.1. OVO**) para obtener la salida, y, con ello, es posible contabilizar los aciertos realizados teniendo en cuenta que se aplica OVO en el proceso de inferencia.

Este proceso se repite con todos los ejemplos del conjunto de entrenamiento. Así, se consigue introducir en la fase de entrenamiento de cada clasificador binario la información del resto de clases, tratando de solventar el problema de los clasificadores no competentes además de conseguir un modelo con una base de datos interpretable, dado que se obtiene esta es única a todas las reglas una vez el entrenamiento del modelo ha concluido.

3.3. FARC-SEMIOVO

Llegados a este punto, surge una nueva variante del modelo *FARC-OVO interpretable*. Consiste en aplicar el algoritmo genético tal como es aplicado en FARC-HD [4], tomando el conjunto BR_{TOTAL} como un clasificador único y olvidando el hecho de que las BR_{c_i} que lo componen provienen de clasificadores diferentes, es decir, prescindiendo de la segunda fase explicada anteriormente. Así, la BR que se optimizará estará formada por la unión de la totalidad de reglas que componen en conjunto BR_{TOTAL} . La diferencia radica en que en este caso el AG no utiliza el proceso de votación de OVO, sino que usa el MRD original de FARC-HD [4].

3.4. Etapa de Filtrado de la Base de Reglas

Mediante el proceso descrito anteriormente, se obtienen dos modelos, *FARC-OVO interpretable* y *FARC-SEMIOVO*, que recuperan la interpretabilidad de la base de datos al ser esta única a la totalidad de reglas. Sin embargo, esta interpretabilidad no se recupera para la base de reglas, ya que al utilizar el método OVO en la fase de extracción, se tienen reglas especializadas en cada subproblema binario. Esto implica que pueda haber reglas redundantes en la base de reglas, además del hecho de que el tamaño de la BR crezca notablemente al extraer una mayor cantidad de reglas en la primera fase. Por ello, se propone aplicar un filtrado que elimina dichas reglas redundantes tratando así de reducir la dimensionalidad de la base de reglas. Este filtrado, se aplica tras la fase de extracción y antes de aplicar el algoritmo evolutivo encargado de su selección.

Para ello, como criterio de selección, se toma, en primer lugar, el peso (confianza) de la regla. En caso de empates, se tomará la multiplicación de la confianza de la regla con su soporte (ejemplos cubiertos por dicha regla). Así, para la regla j -ésima se tendría:

$$c_j = \text{conf}_j * \text{supp}_j$$

Y, si en este caso también se tiene que el valor es el mismo, se utilizará la medida $wWRAracc$ [21] aplicada en la segunda etapa de FARC-HD [4].

El proceso de selección se describe como sigue:

1. Seleccionar un par de reglas de la base de reglas.
2. Comprobar si su antecedente es el mismo.
3. En caso afirmativo, se selecciona aquella regla cuya confianza sea mayor.
4. Si la confianza de ambas reglas es la misma, se selecciona la regla con mayor c_j .
5. En caso de un nuevo empate, seleccionar la regla cuyo $wWRAracc$ sea mayor.
6. Repetir el proceso con cada par de reglas de la base de reglas.

Con esto, se consiguen eliminar de la BR tanto las reglas repetidas, como aquellas que comparten el mismo antecedente, pero cuyo consecuente es distinto.

3.5. Cambio en las Funciones Overlap n-dimensionales

Como se ha visto anteriormente, tanto en FARC-OVO [5] como en este trabajo, se hace uso de funciones overlap [12] para calcular el grado de emparejamiento de un ejemplo con el antecedente de una regla. En estas funciones, descritas en el apartado **2.3. Funciones Overlap**, se tienen en cuenta el número total de variables a la hora de aplicarlas. Sin embargo, en FARC-HD [4] se menciona que las etiquetas utilizadas son de tipo *do not care*, lo que implica que el número de antecedentes de cada regla puede variar y, por tanto, no tiene que ser siempre el mismo que el número total de variables del problema. Esto afecta especialmente a las overlap de la *media geométrica* y *media armónica*. Por ello, se propone una modificación de dichas funciones de manera que únicamente tengan en cuenta el número de antecedentes de cada regla.

Capítulo 4: Marco Experimental

En este capítulo se presenta el contexto elegido para la realización de los experimentos. Primero, se describen los conjuntos de datos (*datasets*) utilizados en el estudio experimental. Después, se muestra la configuración de parámetros considerada en los modelos. Por último, se introducen las medidas de rendimiento y test estadísticos utilizados para analizar los resultados obtenidos.

4.1. Datasets

Con el objetivo de analizar las propuestas, se han considerado veinte conjuntos de datos (*datasets*) del mundo real seleccionados del repositorio de KEEL [28], disponibles públicamente desde el sitio web del proyecto¹, que son los mismo que se utilizan en FARC-OVO [5]. En la *Tabla 1* se resumen las características de los conjuntos seleccionados, indicando por cada *dataset* el número de ejemplos (#Ej.), número de atributos (#Atr.), número de atributos numéricos (#Num.) y nominales (#Nom.), y número de clases (#Clas.).

Para realizar las evaluaciones de los modelos propuestos, se ha considerado un *modelo de validación cruzada de 5 particiones* como el descrito en la sección **2.1. Problemas de Clasificación**. Sin embargo, en lugar de emplear el método habitual de validación cruzada y con el objetivo de corregir el problema de la fractura de los datos (*dataset shift*), es decir, que el conjunto de datos y el conjunto de prueba no siguen la misma distribución [29, 30, 31], se utilizará el llamado *Distribution Optimally Balanced Cross Validation* (DOB-CV) [32].

¹ <https://sci2s.ugr.es/keel/datasets.php>

TABLA 1. DESCRIPCIÓN DE LAS CARACTERÍSTICAS DE LOS DATASET UTILIZADOS EN LA EXPERIMENTACIÓN.

Id.	Dataset	#Ej.	#Atr.	#Num.	#Nom.	#Clas.
aut	autos	159	25	15	10	6
bal	balance	625	4	4	0	3
cle	cleveland	297	13	13	0	5
con	contraceptive	1473	9	6	3	3
eco	ecoli	336	7	7	0	8
gla	glass	214	9	9	0	7
hay	hayes-roth	132	4	4	0	3
iri	iris	150	4	4	0	3
new	newthyroid	215	5	5	0	3
pag	pageblocks	548	10	10	0	5
pen	penbased	1100	16	16	0	10
sat	satimage	643	36	36	0	7
seg	segment	2310	19	19	0	7
shu	shuttle	2175	9	9	0	5
tae	tae	151	5	3	2	3
thy	thyroid	720	21	21	0	3
veh	vehicle	846	18	18	0	4
vow	vowel	990	13	13	0	11
win	wine	178	13	13	0	3
yea	yeast	1484	8	8	0	10

4.2. Configuración de Parámetros y Clasificadores Difusos Utilizados

En esta sección, se muestra la configuración utilizada en la experimentación. Esta, se muestra en la *Tabla 2*, y es la misma que se utiliza en FARC-OVO [5]. Por ello, en los modelos *FARC-OVO interpretable* y *FARC-SEMIOVO* propuestos, se va a utilizar dicha configuración de parámetros. Además, para comparar el rendimiento de los modelos propuestos, se toman los mejores resultados obtenidos en [5] para los modelos FARC-OVO y FARC-HD utilizando funciones de solapamiento, mostrados en la *Tabla 3*.

TABLA 2. CONFIGURACIÓN DE PARÁMETROS.

Algoritmo	Parámetros
FARC-HD	Número de etiquetas lingüísticas por variable: 5 Soporte mínimo: 0.05 Confianza mínima: 0.8 Profundidad máxima: 3 Parámetro k : 2 Número de evaluaciones: 20000 Número de individuos: 50 Parámetro α : 0.02 Bits por gen: 30 Inferencia: Combinación aditiva Número de vecinos: 1 Umbral: 0.05

TABLA 3. MEJORES RESULTADOS DE LOS MODELOS FARC-OVO Y FARC-HD CON FUNCIONES OVERLAP.

Dataset	FARC-OVO ^{HM} _{VOTE}	FARC-HD ^{PROD}
aut	80.99±7.33	80.30±6.79
bal	85.69±3.07	87.51±1.80
cle	57.74±5.31	57.80±3.48
con	55.15±2.31	53.63±2.06
eco	83.20±5.60	82.93±5.00
gla	70.18±4.11	66.28±5.99
hay	81.19±5.98	79.21±5.00
iri	95.33±3.52	95.56±4.08
new	96.28±2.61	95.04±2.70
pag	96.43±1.49	94.18±1.60
pen	94.22±2.49	93.05±2.23
sat	84.05±3.26	80.96±3.60
seg	94.99±1.30	93.62±0.62
shu	99.59±0.22	95.50±1.02
tae	60.55±8.94	57.89±11.35
thy	92.51±0.64	93.52±0.38
veh	71.90±3.05	69.90±2.07
vow	90.71±1.14	75.32±2.38
win	94.55±3.55	95.56±3.99
yea	59.99±3.33	59.60±2.56
AVG.	82.26±3.46	80.37±3.44

4.3. Medidas de Rendimiento y Pruebas Estadísticas

Para comprobar el rendimiento de los modelos, se emplea la métrica más comúnmente utilizada: el grado de precisión o *accuracy rate*, definida como el porcentaje de ejemplos correctamente clasificados en relación con el número total de ejemplos. Además de las medidas de rendimiento, se utilizará el promedio del número de reglas y de antecedentes por regla para medir el efecto en el tamaño de la base de reglas.

También, se van a llevar a cabo algunas pruebas no paramétricas [33, 34] con objeto de otorgar soporte estadístico al análisis de los resultados. Concretamente, se hace uso de la prueba de rangos con signo de Wilcoxon [35] para realizar comparaciones por pares, la prueba de rangos alineados de Friedman [36] para comprobar si existen diferencias estadísticas entre un grupo de métodos, y la prueba de Holm *post-hoc* [37] para averiguar qué algoritmos rechazan la hipótesis nula de equivalencia con respecto al método de control seleccionado. Una descripción más detallada de estas pruebas puede encontrarse en el sitio web <https://sci2s.ugr.es/sicidm>.

Capítulo 5: Estudio Experimental

En este capítulo, se muestran y analizan los resultados obtenidos por los modelos propuestos. Para ello, se ha desarrollado un estudio experimental compuesto por las siguientes etapas:

1. Se presentan los resultados de las propuestas *FARC-OVO interpretable* y *FARC-SEMIOVO* sin añadir las modificaciones propuestas en las funciones de solapamiento, y se analiza si es una solución apropiada para problemas multi-clase en comparación con FARC-HD [4] y FARC-OVO [5].
2. Se analiza el comportamiento de los modelos propuestos introduciendo la modificación propuesta en las funciones de solapamiento (Sección **3.5. Cambio en las Funciones Overlap n -dimensionales**) y se estudia si mejora.
3. Se analiza el impacto de los modelos propuestos en la base de reglas y se comprueba la validez del filtrado propuesto para tratar de añadirle interpretabilidad (Sección **3.4. Etapa de Filtrado de la Base de Reglas**).

5.1. Estudiando los modelos FARC-OVO interpretable y FARC-SEMIOVO sin introducir modificaciones en las funciones de solapamiento

En esta sección, primero se muestran los resultados obtenidos por los modelos *FARC-OVO interpretable* y *FARC-SEMIOVO* sin introducir los cambios mencionados en las funciones de solapamiento, y se analiza el rendimiento de ambos sistemas. A continuación, se mide la bondad de ambos modelos en comparación con los algoritmos FARC-HD [4] y FARC-OVO [5].

La *Tabla 4* muestra el rendimiento obtenido con cada modelo en todos los conjuntos de datos utilizando, por un lado, las funciones de solapamiento mencionadas en la sección **2.3. Funciones Overlap**: PROD, GM, HM y MIN, descartando el *seno* (SIN) debido a la poca precisión obtenida con él en FARC-OVO. Y, por otro, algunas de las funciones antes mencionadas para realizar la agregación del método OVO (**2.7.1. OVO**). En este caso, se toma las funciones de: VOTE, LVPC, WV y WinWV,

5.1.1. Análisis de rendimiento de FARC-OVO interpretable y FARC-SEMIOVO

En la *Tabla 4* se puede observar que tanto en el caso de *FARC-OVO interpretable* como en *FARC-SEMIOVO*, no se aprecian diferencias considerables en el uso de una función de solapamiento u otra. Sin embargo, no se pueden extraer conclusiones significativas sin llevar a cabo el correspondiente análisis estadístico.

Por ello, con objeto de detectar si realmente existen diferencias significativas entre los resultados de cada función de solapamiento, se ha llevado a cabo las pruebas de rangos alineados de Friedman y de Holm post-hoc, cuyos resultados se muestran en la *Tabla 5*. Los resultados están agrupados por columnas de acuerdo con el método utilizado (los diferentes tipos de voto con *FARC-OVO interpretable* por un lado y *FARC-SEMIOVO por otro*), y por filas en función de la función de solapamiento utilizada. Es decir, por cada modelo se comparan las diferentes funciones de solapamiento para ver si alguna de ellas funciona mejor que el resto. El valor de cada celda corresponde al rango obtenido con la prueba de rangos alineados de Friedman, que compara las diferentes funciones de solapamiento por columna, es decir, se ejecutan tantas pruebas de Friedman como columnas (o métodos) haya. Entre paréntesis, se muestra el *p*-valor ajustado obtenido en la prueba de Holm empleando como método de control aquel cuyo rango de Friedman sea menor (mostrado en **negrita**). El *p*-valor ajustado está subrayado cuando existen diferencias estadísticas significativas, utilizando como umbral de confianza $\alpha = 0.1$.

TABLA 4. PRECISIÓN DE LOS MODELOS FARC-OVO INTERPRETABLE Y FARC-SEMIOVO SI CAMBIOS EN LAS FUNCIONES OVERLAP.

Dataset	OVO Interpretable																SEMI-OVO			
	VOTE				LVPC				WEIGHTED				WinWV				PROD	GM	HM	MIN
	PROD	GM	HM	MIN	PROD	GM	HM	MIN	PROD	GM	HM	MIN	PROD	GM	HM	MIN				
aut	80,73	79,85	84,52	82,67	78,69	83,18	76,95	78,79	81,83	78,79	76,40	80,64	75,78	80,67	80,00	77,51	76,30	81,42	77,78	73,81
bal	87,84	88,33	88,96	86,73	89,12	87,53	87,04	86,87	88,32	87,35	87,52	85,74	88,15	87,83	88,15	86,39	87,83	89,12	87,98	88,31
cle	59,31	57,61	55,92	54,55	55,90	56,59	55,93	55,61	59,31	58,97	57,31	54,26	57,93	56,94	55,65	55,22	58,63	56,28	58,63	56,89
con	54,78	54,45	53,83	54,51	54,44	53,97	55,81	55,53	54,52	56,21	54,86	54,10	52,95	53,56	54,72	55,46	54,44	53,70	54,72	55,26
eco	84,35	81,05	81,94	84,05	85,48	82,81	82,91	85,25	85,76	81,12	81,73	84,96	85,28	81,67	82,56	83,71	84,59	84,57	84,60	82,79
gla	55,29	62,26	60,79	53,79	56,65	58,54	52,92	53,36	56,65	58,95	57,02	60,36	56,55	67,03	57,52	58,90	66,87	71,52	70,64	68,81
hay	80,38	80,44	78,85	80,38	78,85	80,38	81,15	78,85	81,81	79,56	81,10	81,81	80,33	82,64	78,90	80,33	80,44	81,92	81,10	80,44
iri	97,33	96,00	96,67	95,33	94,67	96,00	95,33	96,67	94,67	96,00	95,33	96,67	94,67	95,33	96,00	95,33	96,00	95,33	96,00	97,33
new	94,88	92,56	95,35	95,81	92,56	94,42	93,95	94,88	93,95	93,49	94,88	95,81	95,35	95,35	94,88	97,67	94,42	94,42	94,88	95,81
pag	95,64	95,26	94,72	94,35	94,17	95,63	95,81	95,82	95,28	94,91	95,46	94,35	94,17	95,45	95,26	94,34	93,78	94,35	94,91	94,72
pen	94,83	94,29	93,47	94,55	96,01	94,83	94,29	95,83	95,28	94,19	94,83	95,47	95,47	93,83	94,65	95,10	93,10	93,83	93,01	93,28
sat	72,02	74,71	74,02	78,71	72,21	64,28	65,37	68,81	62,21	64,61	65,84	60,67	75,46	79,04	75,93	77,47	84,16	80,26	80,88	82,75
seg	92,94	92,95	93,07	93,59	94,33	93,46	93,64	93,85	94,24	93,46	94,11	93,98	93,68	93,25	93,72	94,29	93,29	91,95	92,81	94,11
shu	97,56	99,68	99,54	99,31	96,14	99,72	99,49	99,68	99,03	99,68	99,49	99,54	99,68	99,63	99,50	99,59	99,63	99,59	99,59	99,59
tae	56,22	62,13	56,10	55,44	54,26	62,11	56,24	54,22	56,15	58,86	62,13	54,13	56,84	57,39	57,51	55,42	58,13	52,79	58,17	57,44
thy	93,89	92,64	92,78	93,20	97,91	92,51	92,51	93,90	97,50	93,90	93,62	94,87	97,50	93,05	94,03	95,57	93,06	95,14	95,98	93,47
veh	72,83	71,28	68,55	70,58	71,40	72,22	72,22	73,76	72,34	69,14	72,34	71,96	72,92	70,68	70,20	69,75	70,79	70,10	70,69	71,04
vow	86,57	83,54	85,86	86,67	89,19	86,06	86,57	88,89	89,19	86,06	86,57	88,89	88,38	85,86	85,56	87,88	82,73	76,36	79,90	81,62
win	96,68	97,78	96,58	95,49	97,16	94,93	96,66	97,23	94,31	94,95	95,58	96,63	95,58	95,54	94,97	96,06	94,95	96,07	96,60	95,47
yea	58,90	59,11	58,91	59,64	59,17	58,84	59,31	61,05	59,98	58,49	59,97	60,65	59,98	59,52	59,58	59,24	59,78	58,90	59,17	59,64
AVG.	80,65	80,80	80,52	80,47	80,42	80,40	79,71	80,44	80,62	79,93	80,31	80,27	80,83	81,21	80,46	80,76	81,15	80,88	81,40	81,13

TABLA 5. PRUEBA DE RANGOS ALINEADOS DE FRIEDMAN Y DE HOLM PARA COMPARAR LAS DIFERENTES FUNCIONES DE SOLAPAMIENTO SIN INTRODUCIR CAMBIOS EN ELLAS.

	OVO ^{VOTE}	OVO ^{LVPC}	OVO ^{WV}	OVO ^{WinWV}	SEMI-OVO
PROD	37,55	39,10 (0.53)	35,55	37,93 (1.00)	42,75 (0.52)
GM	39,53 (1.00)	41,55 (0.53)	46,68 (0.39)	37,05	45,70 (0.38)
HM	44,53 (1.00)	47,98 (0.14)	42,73 (0.66)	46,18 (0.64)	34,48
MIN	40,40 (1.00)	33,38	37,05 (0.84)	40,85 (1.00)	39,08 (0.53)

Se puede observar que para el modelo *FARC-SEMIOVO* (última columna de la *Tabla 5*), si bien la media armónica es la que mejores resultados ofrece, no existen diferencias estadísticas entre las cinco funciones.

En *FARC-OVO interpretable*, para las diferentes funciones de agregación se observa que tampoco existen diferencias significativas entre las cinco funciones de solapamiento, si bien el producto (PROD) se ve como la que mejores resultados ofrece a pesar de que cuando se hace uso de las estrategias LVPC y WinWV las mejores funciones de solapamiento son la del mínimo (MIN) y media geométrica (GM) respectivamente.

5.1.2. Comparativa con FARC-HD y FARC-OVO

A continuación, se quiere comprobar si los modelos propuestos mejoran o no a sus predecesores, FARC-HD y FARC-OVO. El objetivo en este caso es conseguir superar al algoritmo original de FARC-HD y tratar de que las propuestas no sean muy inferiores a FARC-OVO. Para el modelo *FARC-OVO interpretable*, se considera para cada tipo de voto el método de control, es decir, aquel que mejor rango obtiene según la *Tabla 5*, si bien no existen diferencias considerables con el resto de las cinco funciones. Para *FARC-SEMIOVO* también se considera el método de control, es decir, la media armónica.

Se han llevado a cabo diferentes comprobaciones por pares mediante la prueba de rangos con signo de Wilcoxon, con intención de comparar los modelos propuestos con FARC-HD, por un lado, y con FARC-OVO por otro. Al igual que antes, se utiliza como nivel de confianza un $\alpha = 0.1$. Estos resultados se muestran en la *Tabla 6* y en la *Tabla 7* respectivamente.

TABLA 6. PRUEBA DE WILCOXON PARA COMPARAR FARC-HD Y LOS MEJORES MODELOS DE LA PROPUESTA.

Comparativa vs FARC-HD	R+	R-	Hipótesis	p-valor
OVO_{PROD}^{VOTE}	148	62	No Rechazada	0.108
OVO_{MIN}^{LVPC}	129	81	No Rechazada	0.370
OVO_{PROD}^{WV}	142.5	67.5	No Rechazada	0.162
OVO_{GM}^{WinWV}	127.5	82.5	No Rechazada	0.401
SEMI- OVO_{HM}	172	38	Rech. SEMI- OVO_{HM} 90%	0.012

R+ son rangos a favor de las propuestas y R- a favor de FARC-HD

En la *Tabla 6* se comprueba que únicamente el modelo *FARC-SEMIOVO* es estadísticamente mejor que FARC-HD original (su *p*-valor es menor que 0.1). El modelo *FARC-OVO interpretable* en cambio, es incapaz de superarlo con ninguno de los tipos de voto.

TABLA 7. PRUEBA DE WILCOXON PARA COMPARAR FARC-OVO Y LOS MEJORES MODELOS DE LA PROPUESTA.

Comparativa vs FARC- OVO_{HM}^{VOTE}	R+	R-	Hipótesis	p-valor
OVO_{PROD}^{VOTE}	80	130	No Rechazada	0.351
OVO_{MIN}^{LVPC}	80	130	No Rechazada	0.351
OVO_{PROD}^{WV}	89	121	No Rechazada	0.550
OVO_{GM}^{WinWV}	46.5	163.5	Rech. FARC- OVO_{HM}^{VOTE} 90%	0.03
SEMI- OVO_{HM}	74	136	No Rechazada	0.247

R+ son rangos a favor de las propuestas y R- a favor de FARC-OVO

En la *Tabla 7*, se puede ver que; a excepción del WinWV que es peor, el resto de las propuestas obtienen resultados competitivos con FARC-OVO, puesto que no hay diferencias estadísticas. Sin embargo, en estos casos los rangos obtenidos son más pequeños que los de OVO, por lo que se ve que son ligeramente peores.

Se concluye pues que, con esta propuesta, se consigue un modelo que puede decirse que es estadísticamente superior a FARC-HD, el modelo *FARC-SEMIOVO* utilizando la media armónica como función de solapamiento. Sin embargo, su *p*-valor en comparación con FARC-OVO no es muy alto, por lo que no se tienen argumentos suficientes para asegurar que no es muy inferior a FARC-OVO.

SOLAPAMIENTO.

Dataset	OVO																SEMI-OVO			
	VOTE				LVPC				WEIGHTED				WinWV				PROD	GM	HM	MIN
	PROD	GM	HM	MIN	PROD	GM	HM	MIN	PROD	GM	HM	MIN	PROD	GM	HM	MIN				
aut	80,73	77,28	80,07	82,67	78,69	79,89	81,2	78,79	81,83	77,58	82,51	80,64	75,78	76,37	78,17	77,51	76,3	77,68	80,11	73,81
bal	87,84	88,31	87,84	86,73	89,12	87,52	88	86,87	88,32	88	87,83	85,74	88,15	87,84	86,87	86,39	87,83	89,44	88,47	88,31
cle	59,31	57,25	55,59	54,55	55,9	58,28	56,98	55,61	59,31	58,63	59,99	54,26	57,93	60,67	57,32	55,22	58,63	55,91	57,94	56,89
con	54,78	54,72	53,7	54,51	54,44	54,18	56,01	55,53	54,52	54,79	56,01	54,1	52,95	54,72	55,74	55,46	54,44	54,72	53,22	55,26
eco	84,35	83,43	84,02	84,05	85,48	83,16	84,92	85,25	85,76	83,44	84,3	84,96	85,28	81,98	82,59	83,71	84,59	83,15	85,87	82,79
gla	55,29	61,86	58,48	53,79	56,65	57,05	54,76	53,36	56,65	60,8	61,82	60,36	56,55	62,07	61,27	58,9	66,87	68,22	72,01	68,81
hay	80,38	80,44	78,85	80,38	78,85	80,38	81,15	78,85	81,81	79,56	81,1	81,81	80,33	82,64	78,9	80,33	80,44	81,92	81,1	80,44
iri	97,33	94,67	95,33	95,33	94,67	96	95,33	96,67	94,67	96	95,33	96,67	94,67	96	94	95,33	96	95,33	94	97,33
new	94,88	95,35	96,28	95,81	92,56	96,28	95,35	94,88	93,95	96,28	96,74	95,81	95,35	96,74	96,74	97,67	94,42	94,88	95,81	95,81
pag	95,64	94,73	94,52	94,35	94,17	96,19	94,55	95,82	95,28	94,71	95,63	94,35	94,17	95,1	95,1	94,34	93,78	94,37	95,82	94,72
pen	94,83	94,1	94,37	94,55	96,01	95,2	95,19	95,83	95,28	95,46	95,37	95,47	95,47	95,46	95,01	95,1	93,1	93,1	94,36	93,28
sat	72,02	78,88	73,58	78,71	72,21	72,12	69,17	68,81	62,21	64,71	61,74	60,67	75,46	77,81	80,25	77,47	84,16	80,91	84	82,75
seg	92,94	93,2	93,85	93,59	94,33	93,42	93,98	93,85	94,24	93,2	93,81	93,98	93,68	93,12	94,2	94,29	93,29	92,34	93,81	94,11
shu	97,56	97,57	97,15	99,31	96,14	98,12	98,4	99,68	99,03	98,76	98,26	99,54	99,68	98,21	98,62	99,59	99,63	99,36	99,27	99,59
tae	56,22	54,86	59,35	55,44	54,26	61,51	58,88	54,22	56,15	54,21	56,9	54,13	56,84	55,6	56,81	55,42	58,13	56,13	55,39	57,44
thy	93,89	93,75	93,34	93,2	97,91	92,51	92,5	93,9	97,5	93,89	93,34	94,87	97,5	93,89	93,2	95,57	93,06	93,33	93,48	93,47
veh	72,83	70,92	74,34	70,58	71,4	70,91	73,51	73,76	72,34	71,98	74,1	71,96	72,92	71,5	73,75	69,75	70,79	70,33	70,81	71,04
vow	86,57	88,28	89,09	86,67	89,19	87,47	89,7	88,89	89,19	87,47	89,7	88,89	88,38	88,08	89,29	87,88	82,73	83,94	83,94	81,62
win	96,68	96,06	97,22	95,49	97,16	97,27	94,41	97,23	94,31	96,65	96,07	96,63	95,58	94,95	94,47	96,06	94,95	95,58	96,63	95,47
yea	58,9	60,17	59,63	59,64	59,17	60,11	59,1	61,05	59,98	59,17	61	60,65	59,98	59,04	60,99	59,24	59,78	59,09	60,18	59,64
AVG.	80,65	80,79	80,83	80,47	80,42	80,88	80,65	80,44	80,62	80,27	81,08	80,27	80,83	81,09	81,16	80,76	81,15	80,99	81,81	81,13

5.2. Estudiando modelos FARC-OVO interpretable y FARC-SEMIOVO incluyendo modificaciones en las funciones de solapamiento

En esta sección, se realiza el estudio de los modelos *FARC-OVO interpretable* y *FARC-SEMIOVO* incluyendo las modificaciones propuestas en **3.5. Cambio en las Funciones Overlap n -dimensionales**. Al igual que en la sección anterior, primero se muestra la precisión obtenida con ambos modelos para las distintas funciones de solapamiento y tipos de voto y se analiza su rendimiento. Después, se mide la bondad de estos modelos con respecto a los clasificadores FARC-HD [4] y FARC-OVO [5], cuyos resultados se muestran en la *Tabla 3*.

5.2.1. Análisis de rendimiento de FARC-OVO interpretable y FARC-SEMIOVO con las nuevas funciones de solapamiento

En la *Tabla 8* se observar que el cambio propuesto ofrece mejor rendimiento que los obtenidos en la *Tabla 4*, especialmente en el modelo *FARC-SEMIOVO* y *FARC-OVO interpretable* utilizando el WV. Se aprecia que, con este cambio, al utilizar la media armónica los resultados mejoran siempre. Con la media geométrica en cambio se aprecia que no mejora tanto ya que cae ligeramente en algunos casos. Para visualizar mejor este cambio en las funciones overlap, se lleva a cabo la prueba de rangos con signo de Wilcoxon, en la que se comparan los resultados obtenidos para la GM y la HM con y sin el cambio para cada tipo de voto. Estos resultados se muestran en la *Tabla 9* y en la *Tabla 10* respectivamente y se observa que con la media geométrica, al introducir el cambio propuesto, la precisión cae ligeramente tanto con el voto en *FARC-OVO interpretable* como en *FARC-SEMIOVO* y con el resto mejora. Con la media armónica en cambio el rendimiento siempre mejora, de hecho, en *FARC-OVO interpretable* con la LVPC es estadísticamente mejor la nueva implementación que la anterior.

TABLA 9. COMPARATIVA MEDIA GEOMÉTRICA SIN CAMBIO VS CON CAMBIO.

Comparativa sin cambios vs con cambios	R+	R-	Hipótesis	p-valor
OVO_{GM}^{VOTE}	110,5	99,5	No Rechazada	0,872
OVO_{GM}^{LVPC}	82	128	No Rechazada	0,381
OVO_{GM}^{WV}	77,5	132,5	No Rechazada	0,267
OVO_{GM}^{WinWV}	102,5	107,5	No Rechazada	0,936
SEMI-OVO _{GM}	106	104	No Rechazada	0,896

R+ son rangos a favor de las funciones de solapamiento sin el cambio y R- a favor del cambio.

TABLA 10. COMPARATIVA MEDIA HARMÓNICA SIN CAMBIO VS CON CAMBIO.

Comparativa sin cambios vs con cambios	R+	R-	Hipótesis	p- valor
OVO_{HM}^{VOTE}	89,5	120,5	No Rechazada	0,546
OVO_{HM}^{LVPC}	46,5	163,5	Rech. HM con cambio 90%	0,028
OVO_{HM}^{WV}	57,5	152,5	Rech. HM con cambio 90%	0,085
OVO_{HM}^{WinWV}	74,5	135,5	No Rechazada	0,243
SEMI-OVO _{HM}	74,5	135,5	No Rechazada	0,277

R+ son rangos a favor de las funciones de solapamiento sin el cambio y R- a favor del cambio.

Al igual que en la sección anterior, con objeto de detectar si realmente existen diferencias significativas entre los resultados de cada función de solapamiento mostradas en la *Tabla 8* se han llevado a cabo las pruebas de rangos alineados de Friedman y de Holm post-hoc, cuyos valores se muestran en la *Tabla 11*.

TABLA 11. RANGOS ALINEADOS DE FRIEDMAN Y DE HOLM PARA COMPARAR LAS DIFERENTES FUNCIONES DE SOLAPAMIENTO INTRODUCIENDO LOS CAMBIOS PROPUESTOS.

	OVO^{VOTE}	OVO^{LVPC}	OVO^{WV}	OVO^{WinWV}	SEMI-OVO
PROD	37,85	44,03 (1.00)	38,53 (0.28)	41,18 (1.00)	43,99 (<u>0.06</u>)
GM	39,90 (1.00)	36,95	47,69 (<u>0.06</u>)	41,50 (1.00)	48,45 (<u>0.02</u>)
HM	38,50 (1.00)	38,69 (1.00)	30,55	36,60	27,99
MIN	45,75 (0.85)	42,33 (1.00)	45,23 (<u>0.09</u>)	42,73 (1.00)	41,55 (<u>0.07</u>)

En este caso, se puede observar que para el modelo *FARC-SEMIOVO*, el uso de la media armónica es claramente la mejor de las funciones de solapamiento. En el caso de *FARC-OVO interpretable*, es uso de la media armónica también parece la mejor de las opciones. Si bien en el caso del VOTE y la LVPC no es la que mejor rango tiene, en esos casos no existen diferencias considerables entre el uso de una función u otra, por lo que se puede concluir que la media armónica es la mejor de las funciones para OVO.

5.3.2. Comparativa con FARC-HD y FARC-OVO

Al igual que en la sección anterior, a continuación, se comparan los mejores modelos propuestos con FARC-HD original y FARC-OVO empleando la prueba de rangos con signo de Wilcoxon.

En este caso también se utilizará el método de control para cada modelo, es decir, el mejor overlap con cada tipo de voto. Los resultados de la comparativa con FARC-HD se muestran en la *Tabla 12* y con FARC-OVO en la *Tabla 13*.

TABLA 12. PRUEBA DE WILCOXON PARA COMPARAR FARC-HD CON LOS MEJORES MODELOS DE LA PROPUESTA INTRODUCIENDO CAMBIOS EN LAS FUNCIONES DE SOLAPAMIENTO.

Comparativa vs FARC-HD	R+	R-	Hipótesis	p-valor
OVO_{PROD}^{VOTE}	148	62	No Rechazada	0.108
OVO_{GM}^{LVPC}	157.5	52.5	Rech. OVO_{GM}^{LVPC} 90%	0.05
OVO_{HM}^{WV}	162	48	Rech. OVO_{HM}^{WV} 90%	0.03
OVO_{HM}^{WinWV}	121	89	No Rechazada	0.55
SEMI- OVO_{HM}	174	36	Rech. SEMI- OVO_{HM} 90%	0.01

R+ son rangos a favor de las propuestas y R- a favor de FARC-HD.

De acuerdo con los resultados mostrados en la *Tabla 12*, se tienen tres modelos cuyo uso es estadísticamente mejor al de FARC-HD. Estos son el modelo *FARC-SEMIOVO* utilizando la media armónica (HM); y *FARC-OVO interpretable* utilizando la media geométrica y la LVPC, por un lado, y la media armónica y el voto ponderado (WV) por otro. Para el resto de los métodos del modelo *FARC-OVO interpretable* no existen pruebas suficientes para determinar que son estadísticamente superiores a FARC-HD original.

Si se observan los resultados mostrados en la *Tabla 13* para los modelos que son estadísticamente superiores en comparación con FARC-HD, es decir; OVO_{GM}^{LVPC} , OVO_{HM}^{WV} y SEMI- OVO_{HM} , se tiene que no hay argumentos suficientes para decir que FARC-OVO sea estadísticamente superior a los modelos propuestos.

TABLA 13. PRUEBA DE WILCOXON PARA COMPARAR FARC-OVO CON LOS MEJORES MODELOS DE LA PROPUESTA INTRODUCIENDO CAMBIOS EN LAS FUNCIONES DE SOLAPAMIENTO.

Comparativa vs FARC- OVO_{HM}^{VOTE}	R+	R-	Hipótesis	p-valor
OVO_{PROD}^{VOTE}	80	130	No Rechazada	0.351
OVO_{GM}^{LVPC}	71.5	138.5	No Rechazada	0.20
OVO_{HM}^{WV}	116.5	93.5	No Rechazada	0.69
OVO_{HM}^{WinWV}	57	153	Rech. OVO_{HM}^{WinWV} 90%	0.07
SEMI- OVO_{HM}	87	123	No Rechazada	0.51

R+ son rangos a favor de las propuestas y R- a favor de FARC-OVO.

En resumen, se tiene que, al introducir las modificaciones en las funciones de solapamiento, los modelos *FARC-OVO interpretable* utilizando la media armónica y el WV (OVO_{HM}^{WV}) y el modelo *FARC-SEMIOVO* con la media armónica (SEMI- OVO_{HM}) son estadísticamente mejores que FARC-HD original y no se tienen pruebas suficientes para determinar que FARC-OVO sea estadísticamente superior. De acuerdo con las precisiones mostradas en la *Tabla 8*, la propuesta de *FARC-SEMIOVO* parece una buena alternativa por ser la que tiene una mayor precisión en comparación con el resto.

5.3. Efecto de las propuestas en la base de reglas y análisis de la etapa de filtrado

En este apartado se muestra, por un lado, el efecto de las propuestas *FARC-OVO interpretable* y *FARC-SEMIOVO* en la base de reglas, comparándolas con el promedio de reglas y de antecedentes por regla en FARC-HD y FARC-OVO (*Tabla 14*). Por otro, se analiza el efecto de una etapa de filtrado que trata de eliminar las posibles redundancias en la base de reglas surgidas en la fase de extracción (**3.1. Fase 1: Extracción de Reglas Aplicado Técnicas de Descomposición**).

5.3.1. Efecto de FARC-OVO Interpretable y FARC-SEMIOVO en la base de reglas

La *Tabla 14* muestra el promedio del número de reglas y antecedentes por regla obtenido en FARC-HD original y FARC-OVO, así como el número de clasificadores base (#BC) empleados con OVO.

En la *Tabla 15* se puede ver el promedio del número de reglas y de antecedentes por regla de los modelos *FARC-OVO Interpretable* y *FARC-SEMIOVO* tras introducir los cambios en las funciones de solapamiento. En esta se observa que, el número de reglas con respecto a FARC-HD original (*Tabla 14*) crece considerablemente, teniendo que en este último el promedio más alto de de 43.40 reglas mientras que con los modelos propuestos la que menos reglas tiene son 130.

Para hacer una comparativa justa con el modelo FARC-OVO original, hay que multiplicar el número de reglas por cada clasificador base por el número total de clasificadores base de cada problema, ya que en los modelos propuestos se juntan todas las reglas extraídas en la primera fase para cada clasificador base en una única base de reglas. Así, se tiene que en comparación con el modelo FARC-OVO original, el número de reglas es similar, teniendo aproximadamente 130 reglas en los modelos propuestos y, como mínimo 134 para FARC-OVO (*15.55 clasificadores base de media x 8.62 reglas por clasificador en el caso del producto*).

Este crecimiento en la dimensionalidad de la base de reglas con respecto a FARC-HD es debido a que, en la fase de extracción, se utiliza el método de descomposición OVO, y después, se juntan todas las reglas obtenidas en una base de reglas única provocando que en la base de reglas pueda haber redundancias: reglas con el mismo antecedente. Por ello, y con intención de reducir el número de reglas y tratar de hacer la base de reglas más interpretable se ha introducido una fase de filtrado presentada en **3.4. Etapa de Filtrado de la Base de Reglas**.

5.3.2. Fase de filtrado

Tal como se ha mencionado antes, el objetivo de esta fase de filtrado es tratar de reducir la dimensionalidad de la base de reglas eliminando aquellas que son redundantes (mismo antecedente). Este proceso se describe en **3.4. Etapa de Filtrado de la Base de Reglas**. Este filtrado, se aplica únicamente al modelo *FARC-SEMIOVO*, puesto que para *FARC-OVO Interpretable*, dado que se utiliza el método OVO durante la etapa del genético, el eliminar reglas provocaría que el modelo perdiese precisión.

En la *Tabla 16* se observa que este filtrado reduce considerablemente el número de reglas, haciendo que sea más fácil de interpretar. Sin embargo, en la *Tabla 17* se aprecia que esta reducción, afecta negativamente a la precisión del modelo, por lo que no por ahora esta fase no resulta útil. Esto puede ser debido al método de razonamiento difuso (de la suma) que se emplea en el algoritmo.

TABLA 14. PROMEDIO DEL NÚMERO DE REGLAS Y ANTECEDENTES POR REGLA EN FARC-HD Y FARC-OVO.

Dataset	#BC	FARC-HD										OVO									
		Número de reglas					Número de antecedentes por regla					Número de reglas					Número de antecedentes por regla				
		PROD	MIN	HM	GM	SIN	PROD	MIN	HM	GM	SIN	PROD	MIN	HM	GM	SIN	PROD	MIN	HM	GM	SIN
aut	15	38.60	33.00	33.00	33.00	32.00	2.65	2.68	2.67	2.70	2.76	4.75	4.64	4.56	4.56	4.43	1.62	1.58	1.54	1.53	1.59
bal	3	62.40	60.00	55.00	55.80	54.40	1.83	1.83	1.80	1.81	1.82	19.20	20.60	21.33	22.00	22.27	1.49	1.51	1.50	1.50	1.46
cle	10	62.20	60.80	64.00	60.80	64.00	2.90	2.90	2.88	2.89	2.90	13.60	13.40	14.10	14.92	14.60	2.56	2.50	2.56	2.54	2.56
con	3	62.40	91.00	104.20	108.00	117.80	2.70	2.80	2.86	2.89	2.93	50.73	65.20	75.40	78.80	86.13	2.64	2.74	2.81	2.83	2.85
eco	28	30.40	36.00	39.00	43.00	41.60	2.46	2.58	2.64	2.70	2.71	3.95	4.04	4.54	4.83	4.54	1.34	1.40	1.43	1.46	1.46
gla	21	22.20	25.80	30.20	30.20	33.00	2.59	2.72	2.79	2.75	2.73	2.87	2.83	3.44	3.60	3.73	1.14	1.15	1.13	1.12	1.13
hay	3	20.80	20.80	19.80	16.40	16.80	2.16	2.16	2.17	2.04	2.08	9.47	9.47	9.33	9.27	9.13	1.67	1.67	1.66	1.68	1.66
iri	3	4.40	3.60	6.40	6.20	6.80	1.06	1.15	1.45	1.37	1.42	2.67	2.60	2.93	3.33	3.00	1.08	1.13	1.27	1.22	1.19
new	3	10.40	10.00	9.60	9.60	10.20	1.65	1.54	1.81	1.63	1.55	3.40	3.67	3.80	3.80	4.13	1.21	1.23	1.12	1.11	1.12
pag	10	12.40	14.60	14.00	13.80	14.00	2.46	2.45	2.15	2.26	2.45	3.20	2.90	2.88	2.98	3.06	1.32	1.24	1.27	1.29	1.44
pen	45	77.80	81.40	95.40	99.80	101.60	2.80	2.84	2.90	2.91	2.91	4.63	4.71	5.05	4.89	5.02	1.26	1.31	1.33	1.36	1.37
sat	21	18.40	17.80	23.40	25.60	31.00	2.63	2.85	2.83	2.82	2.75	1.97	2.32	3.43	3.72	4.34	1.34	1.32	1.16	1.13	1.10
seg	21	41.60	44.20	62.40	60.80	62.40	2.65	2.70	2.77	2.79	2.77	3.99	4.57	6.25	6.24	6.60	1.33	1.31	1.39	1.41	1.38
shu	10	6.60	5.20	7.00	6.60	6.60	1.90	1.86	2.00	1.95	2.05	2.28	2.18	2.28	2.34	2.26	1.44	1.34	1.31	1.25	1.26
tae	3	18.20	28.60	28.60	30.00	28.00	2.55	2.60	2.80	2.79	2.79	15.13	16.27	17.47	17.93	17.67	2.34	2.45	2.55	2.57	2.57
thy	3	4.40	4.00	4.80	3.60	5.60	2.13	1.89	1.91	1.80	1.79	2.87	3.13	5.93	4.87	5.53	1.78	1.76	1.81	1.74	1.75
veh	6	42.80	52.00	67.20	73.40	78.20	2.65	2.89	2.95	2.93	2.92	13.17	16.90	24.23	26.83	28.50	2.14	2.35	2.53	2.57	2.58
vow	55	71.60	75.40	83.20	84.80	95.20	2.87	2.91	2.95	2.96	2.96	5.51	5.57	6.85	7.41	7.37	1.79	1.85	1.99	2.02	2.01
win	3	8.60	8.20	10.20	10.60	12.20	1.63	1.77	1.80	1.93	2.04	3.47	3.33	3.73	4.13	4.47	1.09	1.08	1.01	1.08	1.22
yea	45	34.00	42.00	48.20	50.20	56.60	2.63	2.73	2.80	2.81	2.88	5.50	5.90	6.80	7.22	7.58	1.64	1.73	1.76	1.77	1.80
AVG.	15.55	32.51	35.72	40.28	41.11	43.40	2.34	2.39	2.45	2.44	2.46	8.62	9.71	11.22	11.68	12.22	1.61	1.63	1.66	1.66	1.67

TABLA 15. PROMEDIO DEL NÚMERO DE REGLAS Y DE ANTECEDENTES POR REGLA EN FARC-OVO INTERPRETABLE Y FARC-SEMI-OVO INTRODUCIENDO LOS PROPUESTOS CAMBIOS EN LAS FUNCIONES OVERLAP.

Dataset	OVO Interpretable								SEMI-OVO							
	Núm. reglas				Núm. de antecedentes por regla				Núm. reglas				Núm. de antecedentes por regla			
	PROD	GM	HM	MIN	PROD	GM	HM	MIN	PROD	GM	HM	MIN	PROD	GM	HM	MIN
aut	95,85	92,25	90,40	92,55	1,79	1,82	1,79	1,75	114,20	106,00	101,80	103,40	1,77	1,83	1,71	1,71
bal	62,50	55,55	59,15	64,50	1,55	1,57	1,54	1,56	72,00	64,40	62,80	71,00	1,49	1,47	1,51	1,50
cle	174,25	171,10	174,90	168,50	2,63	2,61	2,60	2,59	149,80	151,40	146,60	142,20	2,60	2,60	2,56	2,61
con	133,75	184,35	171,05	173,90	2,66	2,70	2,69	2,73	171,20	247,60	243,80	208,20	2,68	2,71	2,70	2,74
eco	167,90	173,45	165,30	167,40	1,45	1,55	1,53	1,52	131,00	151,60	139,20	130,80	1,42	1,50	1,46	1,50
gla	34,20	38,30	38,90	41,60	1,64	1,85	1,75	1,72	72,40	87,00	81,80	83,00	1,66	1,83	1,76	1,69
hay	14,80	15,00	14,05	14,80	1,73	1,70	1,74	1,73	16,80	16,80	16,40	16,80	1,70	1,69	1,67	1,70
iri	4,20	4,90	4,65	4,60	1,18	1,27	1,16	1,14	3,40	5,40	4,60	3,80	1,10	1,36	1,29	1,25
new	8,55	9,85	9,95	8,60	1,28	1,11	1,18	1,17	11,80	12,80	11,40	10,80	1,34	1,18	1,18	1,25
pag	32,80	28,00	28,20	29,00	1,46	1,35	1,38	1,34	37,60	38,80	35,00	33,80	1,30	1,38	1,35	1,33
pen	380,95	417,40	402,65	386,15	1,49	1,68	1,61	1,54	388,80	395,80	400,60	395,00	1,51	1,68	1,62	1,55
sat	17,70	35,75	35,70	23,30	1,68	1,46	1,45	1,64	44,60	81,20	74,60	51,60	1,80	1,46	1,49	1,62
seg	135,75	184,50	166,50	150,40	1,74	1,89	1,77	1,84	109,00	158,00	135,20	120,20	1,72	1,82	1,71	1,78
shu	18,15	24,30	21,35	17,55	1,40	1,42	1,44	1,31	24,20	25,60	24,60	26,40	1,38	1,35	1,43	1,29
tae	28,60	36,05	33,25	34,30	2,37	2,52	2,50	2,54	35,80	44,60	44,80	37,80	2,28	2,52	2,46	2,54
thy	4,55	3,50	4,10	4,80	1,84	1,45	1,75	1,97	6,20	9,80	9,80	9,00	1,87	1,63	1,78	2,01
veh	90,40	137,90	119,50	107,95	2,30	2,66	2,56	2,54	82,00	138,40	125,80	106,80	2,36	2,69	2,59	2,52
vow	531,55	706,45	621,50	527,00	2,13	2,42	2,36	2,21	484,60	556,80	572,00	488,80	2,13	2,39	2,34	2,20
win	10,85	12,00	11,45	10,25	1,14	1,21	1,23	1,14	12,40	13,60	12,40	13,00	1,16	1,27	1,15	1,13
yea	318,45	327,70	318,85	324,00	1,86	1,99	1,95	1,98	289,60	367,20	354,40	312,40	1,87	1,95	1,96	2,00
AVG.	113,29	132,92	124,57	117,56	1,77	1,81	1,80	1,80	112,87	133,64	129,88	118,24	1,76	1,82	1,79	1,80

TABLA 16. PROMEDIO DE NÚMERO DE REGLAS Y NÚMERO DE ANTECEDENTES POR REGLA EN FARC-SEMIOVO CON FILTRADO.

Dataset	SEMI-OVO							
	Número de reglas				Número de antecedentes por regla			
	PROD	GM	HM	MIN	PROD	GM	HM	MIN
aut	63,60	66,60	61,00	61,80	1,97	2,08	2,09	2,05
bal	43,20	44,00	43,00	45,80	1,66	1,67	1,68	1,67
cle	123,00	123,40	124,20	113,80	2,70	2,67	2,67	2,68
con	134,60	199,40	188,40	178,40	2,69	2,72	2,71	2,73
eco	52,60	59,20	57,80	54,20	2,00	2,14	2,18	2,17
gla	42,40	53,40	48,60	46,00	1,99	2,18	2,18	2,12
hay	14,20	14,40	13,60	14,20	1,84	1,85	1,86	1,84
iri	3,20	5,20	3,80	3,60	1,10	1,28	1,25	1,03
new	8,20	8,40	7,40	8,20	1,46	1,24	1,28	1,48
pag	21,00	20,60	19,00	18,40	1,53	1,59	1,67	1,55
pen	163,80	181,80	172,60	159,80	2,04	2,28	2,22	2,15
sat	30,60	44,20	39,20	35,20	1,93	1,75	1,69	1,85
seg	58,40	84,60	79,60	60,00	2,19	2,19	2,23	2,27
shu	12,20	13,60	11,80	11,00	1,67	1,72	1,68	1,59
tae	26,80	36,40	34,80	31,60	2,29	2,51	2,50	2,42
thy	5,40	7,80	6,80	6,40	2,08	1,95	1,92	2,13
veh	76,40	127,60	108,40	87,20	2,39	2,72	2,66	2,61
vow	282,40	348,40	326,80	288,20	2,55	2,69	2,66	2,61
win	10,60	11,60	11,60	10,60	1,11	1,12	1,07	1,12
yea	121,80	152,80	139,80	129,60	2,43	2,59	2,51	2,51
AVG.	64,72	80,17	74,91	68,20	1,98	2,05	2,04	2,03

TABLA 17. PRECISIÓN DEL MODELO FARC-SEMIOVO CON FILTRADO.

Dataset	SEMI-OVO			
	PROD	GM	HM	MIN
aut	76,77	79,43	74,58	74,41
bal	87,03	88,00	87,52	87,04
cle	59,26	56,59	54,94	58,26
con	54,65	54,44	55,40	54,93
eco	78,07	79,95	79,56	78,20
gla	67,32	69,70	72,11	70,11
hay	77,31	75,99	76,65	77,31
iri	96,00	96,00	96,67	94,67
new	95,35	95,81	94,42	95,81
pag	94,19	94,72	95,10	94,74
pen	88,75	91,10	88,37	88,11
sat	83,70	80,09	79,96	82,59
seg	90,35	88,66	91,00	91,91
shu	98,16	99,40	99,03	95,49
tae	53,41	57,46	58,08	55,51
thy	94,86	93,33	93,34	93,34
veh	71,77	71,62	69,49	72,33
vow	75,25	74,75	79,29	76,67
win	95,52	97,24	96,11	96,09
yea	57,22	58,63	56,81	57,83
AVG.	64,72	80,17	74,91	68,2

Capítulo 6: Conclusiones

En este proyecto se han presentado dos métodos para, por un lado, recuperar la interpretabilidad perdida en FARC-HD [4] al introducir OVO, y por otro, tratar de afrontar el problema de los clasificadores no competentes que surge al utilizar la técnica de descomposición OVO [6]. A estos, se les ha denominado *FARC-OVO Interpretable* y *FARC-SEMIOVO*.

Para afrontar los problemas antes mencionados, la solución propuesta ha consistido en aplicar el algoritmo genético de manera global a la totalidad de las reglas extraídas mediante la técnica OVO. Con esto, se consigue una base de datos que es común a la totalidad de las reglas y por tanto al problema que se trate. Como consecuencia, se ha conseguido proporcionar un modelo que es estadísticamente mejor que FARC-HD y cuyo rendimiento no es muy inferior a FARC-OVO, logrando además una base de datos y de reglas únicas.

Sin embargo, se ha mostrado que, en estos modelos, si bien la base de datos es interpretable, la dimensionalidad en la base de reglas crece considerablemente con respecto al algoritmo original de FARC-HD, por lo que no puede concluirse que esta última sea del todo interpretable. Para tratar de solucionar este problema, se propone también una etapa de filtrado para eliminar las reglas redundantes. Esta, es aplicada en *FARC-SEMIOVO*, puesto que para *FARC-OVO interpretable*, el eliminar reglas supondría una pérdida en la precisión. Los resultados obtenidos tras aplicar esta etapa en *FARC-SEMIOVO*, si bien conseguía reducir de manera notable el tamaño de la base de reglas, perdía la precisión obtenida anteriormente, por lo que se concluye que por ahora esta fase es viable. Esto, puede deberse a que el MRD utilizado en el algoritmo es el de la suma, por lo que el eliminar reglas provoca que el algoritmo se pueda confundir.

En resumen, tal como se muestra en el estudio experimental, se consigue superar con algunos de los métodos propuestos a FARC-HD original, siendo su uso estadísticamente superior y están cerca del modelo FARC-OVO. Además, se recupera una parte de la interpretabilidad original de FARC-HD que se perdió al introducir el método OVO con la base de datos.

Capítulo 7: Líneas Futuras

A continuación, se proponen nuevas ideas para tratar de mejorar la propuesta actual:

Con motivo de reducir la dimensionalidad de la base de reglas, se podrá mejorar el filtro propuesto anteriormente, de manera que se consiga reducir el número de reglas sin perder precisión en el sistema.

Otra manera de reducir la dimensionalidad de la base de reglas es tratar de modificar la función fitness de evolutivo para que se tenga en cuenta también el número de reglas seleccionadas. Relacionado con ello, también puede ser interesante probar diferentes tipos de funciones fitness en el algoritmo evolutivo, con motivo de ver cómo esto afecta al rendimiento final del modelo y ver si se consiguen mejorar los resultados.

Por último, en lugar de la propuesta realizada en este TFG basada en la aplicación del algoritmo evolutivo de forma global, tratar de resolverlo valiéndose de los llamados algoritmos co evolutivos es otra posibilidad para futuros trabajos.

Bibliografía

- [1] I. A. Witten, E. Frank y M. A. Hall, Data Mining. Practical machine learning tools and techniques, Morgan Kaufman Publishers, 2011.
- [2] O. Loistl, «Emergence of macro-variables by evaluation and clustering of micro-activities,» 2014.
- [3] A. Simpao, L. Ahumada, J. Gálvez y M. Rehman, «A review of analytics and clinical informatics in health care,» *Journal of Medical Systems*, vol. 38, nº 4, 2014.
- [4] J. Alcalá-Fdez, R. Alcalá y F. Herrera, «A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning.,» *IEEE Transactions on Fuzzy Systems*, vol. 19, nº 5, pp. 857-872, 2011.
- [5] M. Elkano, M. Galar, J. A. Sanz, A. Fernández, E. Barrenechea, F. Herrera y H. Bustince, «Enhancing Multiclass 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.
- [6] M. Galar, A. Fernández, E. Barrenechea, H. Bustince y F. Herrera, «Dynamic classifier selection for one-vs-one strategy: Avoiding non-competent classifiers,» *Pattern Recognition*, vol. 46, nº 12, pp. 3412-3424, 2013.
- [7] A. Lorena, A. Carvalho y J. Gama, «A review on the combination of binary classifiers in multiclass problems,» *Artificial Intelligence Review*, vol. 30, nº 1-4, pp. 19-37, 2008.
- [8] R. Anand, K. Mehrotra, C. Mohan y S. Ranka, «Efficient classification for multiclass problems using modular neural networks,» *IEEE Transactions on Neural Networks*, vol. 6, nº 1, pp. 117-124, 1995.
- [9] P. Clark y R. Boswell, «Rule induction with cn2: some recent improvements,» *EWSL'91: Proceedings of the European Working Session on Machine Learning*, pp. 151-163, 1991.
- [10] M. Galar, A. Fernández, E. Barrenechea, H. Bustince y 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.
- [11] L. A. Zadeh, «Fuzzy Sets,» *Information and Control*, vol. 8, nº 3, pp. 338-353, 1965.

- [12] H. Bustince, J. Fernandez, R. Mesiar, J. Montero y R. Orduna, «Overlap functions,» *Nonlinear Analysis: Theory, Methods & Applications*, vol. 72, nº 3-4, pp. 1488-1499, 2010.
- [13] D. Paternain, M. Pagola, J. Fernandez, R. Mesiar, G. Beliakov y H. Bustince, «Brain MRI thresholding using incomparability and overlap functions,» *Intelligent Systems Design and Applications (ISDA)*, vol. 2011 11th International Conference, pp. 808-812, 2011.
- [14] A. Jurio, H. Bustince, M. Pagola y R. R. Yager, «Some properties of overlap and grouping functions and their application to image thresholding,» *Fuzzy Sets and Systems*, vol. 229, pp. 69-90, 2013.
- [15] D. E. Goldberg, *Genetic algorithms in search, optimization and machine learning*, 1989.
- [16] M. Mitchell, *Introduction to Genetic Algorithms*, 1998.
- [17] G. Beliakov, A. Pradera y T. Clavo, *Aggregation Functions: A Guide for Practitioners*, vol. 221 of *Studies in Fuzziness and Soft Computing*, Springer Berlin Heidelberg, 2007.
- [18] O. Cordón, M. del Jesus y F. Herrera, «A proposal on reasoning methods in fuzzy rule-based classification systems,» *International Journal of Approximate Reasoning*, vol. 20, nº 1, pp. 21-45, 1999.
- [19] H. Ishibuchi y T. Yamamoto, «Rule weight specification in fuzzy rule-based classification systems,» *IEEE Transactions on Fuzzy Systems*, vol. 13, nº 4, pp. 428-435, 2005.
- [20] R. Agrawal y R. Srikant, «Fast algorithms for mining association rules,» de *International Conference on Very Large Data Bases*, Santiago de Chile, Chile, 1994.
- [21] B. Kavsek y N. Lavrac, «Apriori-sd: Adapting association rule learning to subgroup discovery,» *Applied Artificial Intelligence*, vol. 20, nº 7, pp. 543-583, 2006.
- [22] L. Eshelman, «The chc adaptative search algorithm: How to have safe search when engaging in nontraditional genetic recombination,» de *Foundations of Genetic Algorithms*, G. Rawling, Ed. Morgan Kaufmann, 1991, pp. 265-283.
- [23] L. Eshelman y J. Schaffer, «Real-coded genetic algorithms and interval schemata,» de *Foundations of Genetic Algorithms*, vol. 2, D. Whitley, Ed. Morgan Kaufmann, 1993, pp. 187-202.
- [24] M. Lozano, F. Herrera, N. Krasnogor y D. Molina, «Real-coded memetic algorithms with crossover hill-climbing,» *Evolutionary Computation*, vol. 12, nº 3, pp. 273-302, 2004.
- [25] J. Friedman, «Another approach to polychotomous classification,» 1996.

- [26] E. Hüllermeier y 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.
- [27] A. Fernández, M. Calderón, E. Barrenechea, H. Bustince y F. Herrera, «Solving multiclass problems with linguistic fuzzy rule based classification systems based on pairwise learning and preference relations,» *Fuzzy Sets and Systems*, vol. 161, nº 23, pp. 3064-3080, 2010.
- [28] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez y 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*, Vols. %1 de %217:2-3, pp. 255-287, 2011.
- [29] J. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. Chawla y F. Herrera, «A unifying view on dataset shift in classification,» *Pattern Recognition*, vol. 45, nº 1, pp. 521-530, 2012.
- [30] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer y N. D. Lawrence, «Dataset Shift in Machine Learning,» *The MIT Press*, 2009.
- [31] V. López, A. Fernández y H. F., «On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed,» *Informaton Sciences*, vol. 257, pp. 1-13, 2014.
- [32] J. Moreno-Torres, J. Saez y F. Herrera, «Study on the impact of partition-induced dataset shift on k-fold cross-validation,» *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, nº 8, pp. 1304-1312, 2012.
- [33] D. Sheskin, *Handbook of parametric and nonparametric statistical procedures*, Chapman & Hall/CRC, 2006.
- [34] S. García, A. Fernández, L. J. y F. Herrera, «A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability,» *Soft Comput*, vol. 13, nº 10, pp. 959-977, 2009.
- [35] F. Wilcoxon, «Individual comparisons by ranking methods,» *Biometrics*, vol. 1, nº 6, pp. 80-83, 1945.
- [36] J. L. Hodges y E. L. Lehmann, «Ranks methods for combination of independent experiments in analysis of variance,» *Ann. Math. Statist.*, vol. 33, pp. 482-497, 1962.
- [37] S. Holm, «A simple sequentially rejective multiple test procedure,» *Scandinavian Journal of Statistics*, vol. 6, pp. 65-70, 1979.