

# Diseño de un Sistema de Clasificación Basado en Reglas Difusas para *one-class* mediante el aprendizaje genético de la base de datos

Pedro Villar<sup>1</sup>, Ana María Sánchez<sup>1</sup>, Rosana Montes<sup>1</sup>, Francisco Herrera<sup>2</sup>

<sup>1</sup>Dpto. Lenguajes y Sistemas Informáticos. Universidad de Granada, {pvillarc,amlopez,rosana}@ugr.es

<sup>2</sup>Dpto. Ciencias de la Computación e Inteligencia Artificial. Universidad de Granada, herrera@decsai.ugr.es

## Resumen

En este trabajo se propone un Algoritmo Genético para aprender un Sistema de Clasificación Basado en Reglas Difusas para clasificación con una única clase o *one-class classification*, una situación que está presente en numerosos problemas reales en la que los ejemplos disponibles para el aprendizaje son todos de una misma clase (clase objetivo). El Algoritmo Genético aprende la granularidad y las particiones difusas utilizando una función de escala no lineal.

**Palabras Clave:** *One-Class Classification*, Sistemas de Clasificación Basados en Reglas Difusas, Algoritmos Genéticos.

## 1 Introducción

Dentro del aprendizaje automático, los métodos tradicionales de clasificación intentan clasificar un objeto desconocido dentro de un conjunto finito y discreto de categorías (o clases). En el paradigma de *one-class classification* [11], una de las clases está suficientemente caracterizada en los ejemplos del conjunto de entrenamiento y se suele denominar clase positiva o clase objetivo (*target*). Para el resto de clases, que caracterizan el concepto negativo (o simplemente la no pertenencia a la clase objetivo) o no hay ejemplos disponibles, o hay tan pocos ejemplos que no son representativos de ese concepto negativo.

Un Sistema de Clasificación Basado en Reglas Difusas (SCBRD) tiene dos componentes principales: un Sistema de Inferencia y una Base de Conocimiento (BC). En un SCBRD lingüístico, la BC está compuesta de la Base de Reglas (BR), constituida por el conjunto de reglas difusas y la Base de Datos (BD), que contiene las funciones de pertenencia de las particiones difusas asociadas a las variables de entrada.

Dentro de la literatura especializada, hay numerosos métodos para el aprendizaje automático de la BC de un SCBRD, tanto para el aprendizaje completo de la BC como para el aprendizaje o ajuste de alguna de sus partes, pero la mayoría están diseñados para problemas multi-clase, especialmente en el caso del aprendizaje de la BR. Por eso, dentro de nuestro método de aprendizaje de la BC proponemos un nuevo método de aprendizaje de reglas difusas específico para *one-class*, que intenta obtener un conjunto reducido de reglas que caractericen bien a la clase objetivo y utilizaremos un sistema de razonamiento difuso diseñado para *one-class*, que nos permite determinar si un nuevo ejemplo desconocido pertenece o no a la clase objetivo.

En un trabajo previo [14], se proponía un método de clasificación para *one-class* utilizando los SCBRDs, donde un Algoritmo Genético (AG) se utilizaba para el aprendizaje del número de etiquetas difusas (granularidad) de cada variable, manteniendo particiones difusas uniformes. En el método presentado en este trabajo, el AG incluye también la posibilidad de ajustar el dominio de las variables y un mecanismo de aprendizaje de las funciones de pertenencia donde las particiones difusas utilizan una función de escala no lineal que permite definir áreas con mayor “sensibilidad” dentro del dominio. El hecho de añadir el aprendizaje de las particiones difusas permite obtener SBRDCs más compactos (con menos reglas) por lo que resultan mucho más interpretables. El método aprende la BC al completo (BD y BR), usando el AG para el aprendizaje de la BD, mientras que para la generación de la BR se utiliza el algoritmo de aprendizaje de reglas específico para *one-class* mencionado anteriormente.

Para el estudio experimental nos hemos basado en trabajos del paradigma *one-class* [4, 13], que utilizan conjuntos de datos del repositorio UCI [2], que se convierten en problemas de *one-class* al considerar todos los ejemplos de una clase como ejemplos de la clase anómala (*outlier*), eliminándolos del conjunto de entrenamiento y pasándolos al conjunto de datos de test.

El trabajo está estructurado de la siguiente manera. Prime-

ro, en la Sección 2 se realiza una breve introducción al paradigma de *one-class classification*, resaltando sus diferencias frente a la clasificación multi-clase convencional. A continuación, en la Sección 3 se detalla el algoritmo de generación de reglas difusas junto con el método usado para clasificar los ejemplos y en la Sección 4 presentamos los aspectos fundamentales del AG que se propone. La Sección 5 muestra el estudio experimental realizado. Finalmente, las conclusiones de este trabajo se exponen en la Sección 6.

## 2 Clasificación con una única clase

El término *one-class classification* fue propuesto por primera vez en [9], aunque varios autores han utilizado otros términos para designar paradigmas similares, como son detección de anomalías (*outlayer detection* o *anomaly detection*), detección de novedades (*novelty detection*) o aprendizaje de conceptos (*concept learning*). Esa variación de términos tiene que ver más con las aplicaciones en las que se ha utilizado el paradigma *one-class* que con diferencias en los modelos utilizados o en la forma de diseñarlos.

Dentro de la tradicional clasificación multi-clase, los métodos de aprendizaje automático se encuentran con una serie de problemas como la reducción de la tasa de error de clasificación, la capacidad de generalización del modelo, la complejidad del mismo o el tratamiento de la dimensionalidad. Dentro del paradigma *one-class* esos problemas siguen apareciendo y, en algunos casos, son más difíciles de abordar. Por ejemplo, determinar una frontera entre los datos que nos permita separar las clases es más difícil cuando sólo se disponen ejemplos de una clase, ya que no tenemos información sobre la "distancia" a la que se encuentran los ejemplos anómalos en cada dirección y es difícil decidir cómo de ajustada a los ejemplos positivos debe estar esa frontera. Por tanto, es de esperar que se requieran más ejemplos para el aprendizaje que en una clasificación convencional. Una revisión de trabajos de *one-class* se puede encontrar en [7], junto con una taxonomía que distingue tres categorías para agrupar los trabajos publicados:

- Disponibilidad de ejemplos para el conjunto de entrenamiento: se distinguen dos posibilidades, que sólo se disponga de ejemplos de la clase positiva (incluyendo o no muy pocos ejemplos de la clase anómala) o que también se disponga de ejemplos sin clasificar.
- Metodología/algoritmos: en este caso se distinguen los algoritmos basados en máquinas vectores de soporte para *one-class* (*One-Class Support Vector Machines*, OCSVM), paradigma seguido por muchas propuestas en la literatura especializada, y el resto de algoritmos (no basados en OCSVM).
- Dominio de aplicación: donde se distingue el campo de clasificación de textos/documentos (ámbito muy

representado dentro de los trabajos de *one-class*) y otros ámbitos de aplicación.

Debido a la gran cantidad de propuestas basadas en máquinas vectores de soporte, hemos seleccionado una implementación clásica de OCSVM [10] como algoritmo para comparar el rendimiento de nuestra propuesta.

Algunas propuestas de *one-class* presentes en la literatura especializada [4, 13] utilizan conjuntos de datos multi-clase como *benchmarks* para validar los resultados de sus propuestas. La forma usual de actuar supone convertir a dichos conjuntos de datos en problemas binarios (con dos clases), agrupando todas los ejemplos de todas las categorías excepto una dentro de una clase y los ejemplos de la categoría restante en la otra clase. A continuación, se divide el conjunto de datos total en un conjunto de datos de entrenamiento y otro de test, de forma que el conjunto de datos de entrenamiento sólo dispone de un porcentaje de ejemplos representativos de una de las dos nuevas clases (clase objetivo) y el conjunto de datos de test lo constituye el resto de ejemplos de la clase objetivo y todos los ejemplos representativos de la otra clase (clase anómala). Dentro de este esquema se distinguen dos posibilidades:

- Los ejemplos de la categoría que contiene una única clase del conjunto original constituyen el conjunto de datos de la clase positiva mientras que los ejemplos del resto de categorías caracterizarían a la clase anómala, lo que se corresponde con el paradigma de aprendizaje de conceptos.
- Los ejemplos de la categoría que contiene una única clase del conjunto original constituyen la clase anómala mientras que los ejemplos del resto de categorías serían los representantes de la clase objetivo, que se correspondería con el paradigma de detección de anomalías. Esta opción es la que hemos seguido en nuestro trabajo utilizando tres conjuntos de datos del repositorio UCI, usados en algunos de los trabajos que siguen este procedimiento [4, 13].

Obviamente, para el caso de conjunto de datos binarios (dos clases), ambos enfoques son el mismo y sólo cabría hacer consideraciones sobre la diferencia entre escoger una clase u otra como clase positiva en el caso de que el conjunto de datos fuese no balanceado.

La mayoría de propuestas para aprendizaje automático de clasificadores multi-clase utilizan alguna medida de precisión del modelo como el porcentaje de ejemplos bien clasificados. Sin embargo, cuando no se dispone de ejemplos de una clase para entrenar el modelo, ese tipo de medidas pueden llevar a conclusiones erróneas cuando se trabaja con el conjunto de datos de test, ya que es necesario encontrar un equilibrio entre el porcentaje de acierto en la clasificación de ejemplos de la clase positiva (único valor de este tipo

que se podría aplicar dentro del proceso de aprendizaje) y el porcentaje de acierto sobre ejemplos de clase anómala (que sí están disponibles en el conjunto de datos de test). En varios trabajos que siguen el mismo esquema de esta propuesta [4, 13] se utiliza para el conjunto de datos de test la medida denominada área bajo la curva o *Area Under the Curve* (AUC). En este trabajo utilizaremos, aparte del AUC, los porcentajes de clasificación correcta para el análisis de los resultados experimentales.

### 3 Generación de la BR y clasificación de nuevos ejemplos

En esta sección describiremos el algoritmo que se ha usado para generar la BR, propuesto en [14], que está diseñado específicamente para el paradigma *one-class*, además de detallar el método de razonamiento difuso que, una vez determinada la BR, decide si un nuevo ejemplo pertenece a la clase positiva.

#### 3.1 Algoritmo de generación de las reglas difusas

La BR está compuesta por un conjunto de reglas difusas que, en este trabajo, tienen la siguiente estructura para un problema con  $N$  variables:

$$R_k : \text{If } X_1 \text{ is } A_1^k \text{ and } \dots \text{ and } X_N \text{ is } A_N^k \text{ then } Y \text{ is } C$$

donde  $X_1, \dots, X_N$  son las variables consideradas en el problema y  $A_1^k, \dots, A_N^k$  son las etiquetas lingüísticas que representan los valores que toma cada variable. En este tipo de reglas, el antecedente representa un subespacio del espacio completo de búsqueda y el consecuente es la etiqueta de la clase positiva del problema ( $C$ ).

El método de generación de reglas intenta generar un conjunto de reglas compacto, que caracterice de forma ajustada a los ejemplos de la clase positiva del conjunto de entrenamiento mediante la selección de reglas que agrupen a muchos ejemplos. El algoritmo de generación del conjunto de reglas difusas que constituyen la BR sólo decide cuáles son los antecedentes de las reglas (ya que el consecuente sería el mismo para todos, la clase positiva) y su funcionamiento se describe a continuación:

Para cada uno de los ejemplos del conjunto de entrenamiento (todos representativos de la clase positiva) se realizan las acciones siguientes:

- Seleccionar el mejor antecedente posible para ese ejemplo, buscando la etiqueta de cada variable con máximo grado de pertenencia difuso para la componente de dicha variable en el ejemplo.
- Comprobar si esa combinación de antecedentes se ha explorado anteriormente con otro ejemplo. Si no ha

sido así, se decide si se acepta o no la regla correspondiente en base al valor de dos medidas calculadas sobre dicha regla:

- **Media\_cubrimiento**, que es la media de los grados de emparejamiento de todos los ejemplos del conjunto de entrenamiento cuyo grado de emparejamiento con dicha regla sea mayor que cero. Para calcular el grado de emparejamiento de un ejemplo con el antecedente de la regla se utiliza la t-norma del producto.
- **Soporte**, que es la fracción de ejemplos cubiertos por la regla (con grado de emparejamiento mayor que cero) respecto del total de ejemplos de entrenamiento.

El propósito de utilizar estas dos medidas es generar un número compacto de reglas con elevado valor medio en el cubrimiento de los ejemplos que emparejan con ellas y que soporten a muchos ejemplos, por lo que un antecedente explorado se aceptará como regla de la BR si cumple los siguientes dos requisitos:

- $Media\_cubrimiento > umbral\_cubrimiento$
- $Soporte > umbral\_soporte$

Siendo  $umbral\_cubrimiento$  y  $umbral\_soporte$ , parámetros del método con valores en el intervalo  $[0, 1]$ .

#### 3.2 Mecanismo de razonamiento difuso para clasificar nuevos ejemplos

Una vez determinada la BR, el mecanismo de inferencia que decide si un ejemplo nuevo se clasifica o no como perteneciente a la clase positiva consta de dos pasos:

- Se seleccionan todas las reglas con grado de emparejamiento mayor que cero para ese ejemplo (considerando la t-norma del producto) y se calcula el valor medio de dichos grados de emparejamiento.
- Si dicho valor medio es mayor o igual que el parámetro  $umbral\_regla$  (con valores en el intervalo  $[0, 1]$ ), el ejemplo se clasifica como perteneciente a la clase positiva. En caso contrario, se considera un ejemplo "anómalo".

### 4 Algoritmo Genético para aprendizaje de la granularidad de la BD

En este apartado se propone un método de aprendizaje automático de la BC de un SCBRD mediante un AG generacional que permite definir el dominio, la granularidad y la partición difusa de cada variable. El proceso de evaluación de las posibles soluciones incluye el método de generación de la BR y de razonamiento difuso explicado en la sección anterior. Notaremos nuestra propuesta como AG-Gr-P-OC

(Algoritmo Genético de aprendizaje de la Granularidad y Particiones para *One-Class*). Dicho método trata de mejorar la interpretabilidad del método presentado en [14], donde el AG únicamente aprendía la granularidad y las particiones difusas se generaban de forma uniforme.

El AG propuesto es una adaptación del método propuesto en [5] para regresión que considera funciones difusas triangulares para representar las etiquetas. En dicho método se codifican en el cromosoma los límites del universo de discurso de las variables (que pueden ampliarse hasta un 25%), lo que permite que las etiquetas más situadas a los extremos de la partición se puedan ajustar mejor a las características del problema. Además, en dicho método se utiliza una función de escala no lineal que permite definir áreas con mayor “sensibilidad” dentro del dominio de las variables, es decir, áreas de la partición difusa donde las etiquetas están más concentradas y permiten discriminar mejor. Como todos los componentes de la BD se van a adaptar mediante un AG, no es conveniente que el espacio de búsqueda sea muy extenso. Por tanto, se deben usar funciones de escala no lineales parametrizadas con un reducido número de parámetros. Usaremos la función propuesta en [5], que sólo tiene un parámetro  $a$  de “sensibilidad” ( $a \in \mathbb{R}, a > 0$ ):

$$f: [-1, 1] \rightarrow [-1, 1] \quad f(x) = \text{signo}(x) \cdot |x|^a$$

El resultado final es una partición en el intervalo  $[-1, 1]$ , donde el parámetro  $a$  puede originar una partición uniforme ( $a = 1$ ), mayor sensibilidad para los valores centrales ( $a > 1$ ) o mayor sensibilidad para los valores extremos ( $a < 1$ ). La función de escala no lineal sólo se aplica a los tres puntos que definen las etiquetas triangulares en una partición uniforme. La Figura 1 muestra esas opciones para una partición de nueve etiquetas.

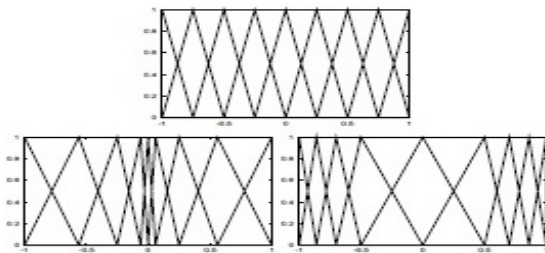


Figura 1: Particiones difusas con  $a = 1$  (arriba),  $a > 1$  (izquierda) y  $a < 1$  (derecha)

Esta función de escala es válida para variables simétricas ya que produce los mismos efectos en la partición difusa de forma simétrica sobre el centro de dicha partición. Para permitir que el área de mayor sensibilidad de la partición se desplace sobre un único extremo de la misma, se añade un nuevo parámetro ( $S$ ) como se describe también en [5]. Dicho parámetro no aumenta significativamente la complejidad del espacio de búsqueda ya que únicamente puede to-

mar dos valores ( $\{0, 1\}$ ), que permiten desplazar el área de mayor sensibilidad hacia un extremo u otro si  $S = 1$  tal y como se puede ver en la Figura 2. Si  $S = 0$  no se produce desplazamiento y las particiones quedarían como en la figura 1.

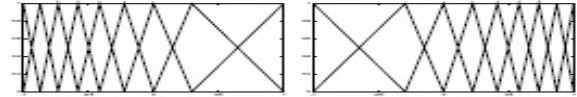


Figura 2: Particiones difusas con  $S = 1$  (izquierda con  $a > 1$  y derecha con  $a < 1$ )

Como los puntos que definen cada una de las funciones de pertenencia se calculan a partir de una partición difusa uniforme, lo único que hace falta para construir la partición de una variable es su dominio, su granularidad y los dos parámetros de “sensibilidad” ( $a$  y  $S$ ). Los siguientes apartados describen los principales componentes de AG-Gr-P-OC.

#### 4.1 Codificación de las soluciones

Cada cromosoma está compuesto de las cuatro partes necesarias para construir la partición difusa de cada variable. Para un problema de clasificación de  $N$  variables de entrada, el dominio se codifica en un vector de números reales de longitud  $2N$ , el número de etiquetas se codifica en un vector entero de longitud  $N$  (los posibles valores considerados son el conjunto  $\{2 \dots 7\}$ ), los parámetros  $a$  de cada variable se representan en un vector de números reales de longitud  $N$  y se utiliza un vector binario de longitud  $N$  para los parámetros  $S$ .

#### 4.2 Población inicial

La población inicial está dividida en tres partes. El primer grupo tiene  $5g$  individuos, siendo  $g$  el número de posibles valores para la granularidad (en nuestro caso,  $g = 6$ ) y todos esos individuos tienen los dominios originales y el mismo número de etiquetas en todas sus variables. Para cada valor posible de granularidad se generan cinco individuos, uno con  $a = 1$ , dos con  $a < 1$  (uno con  $S = 0$  y otro con  $S = 1$ ) y dos con  $a > 1$  (uno con  $S = 0$  y otro con  $S = 1$ ). El segundo grupo tiene una estructura igual al primero excepto en los dominios, que se generan aleatoriamente dentro de los límites establecidos para ellos. El tercer grupo contiene el resto de cromosomas y su inicialización es aleatoria en todos sus componentes.

#### 4.3 Evaluación de los cromosomas

La evaluación de un cromosoma consta de varias etapas:

1. Generar la BD utilizando la información del cromosoma. Para cada variable se construye una partición difusa de acuerdo al dominio, número de etiquetas y parámetros de “sensibilidad”.

2. Generar la BR a partir de la BD mediante el proceso de generación de reglas descrito en la Sección .
3. De acuerdo al mecanismo de inferencia descrito en la sección , clasificar todos los ejemplos del conjunto de aprendizaje. El valor de la función de evaluación será el número de ejemplos del conjunto de entrenamiento correctamente clasificados. El AG evoluciona para maximizar dicho valor a lo largo de las generaciones.

#### 4.4 Operadores genéticos

El mecanismo de selección empleado ha sido el Muestreo Universal Estocástico, propuesto por Baker en [3], incorporando una selección elitista.

El operador de cruce varía en función de la igualdad de los padres en el número de etiquetas de cada variable. Si ambos cromosomas tienen la misma granularidad, se supone que el AG ha encontrado una zona prometedora dentro del espacio de búsqueda que conviene explotar. Por eso, se aplica el cruce max-min-aritmético (MMA) [6] sobre la parte de codificación real del cromosoma, es decir, los dominios y los parámetros  $a$  manteniendo los mismos valores de granularidad en los descendientes. Se generan los cromosomas hijos de ese tipo de cruce con los dos posibles valores para el parámetro  $S$  y se seleccionan los dos mejor adaptados para sustituir a los padres. Si los padres tienen alguna diferencia en la granularidad de alguna de las variables, se escoge aleatoriamente un punto de cruce  $p$  (entre el rango de valores  $\{2 \dots N\}$ ) y los dos padres se cruzan sobre la variable  $p$  en cada una de las cuatro partes del cromosoma.

El operador de mutación dependerá de la parte del cromosoma en que se encuentra el gen que debe mutar. Si es un valor de granularidad, el operador es el propuesto por Thrift en [12]. En el gen al que corresponde mutar se cambia el valor de número de etiquetas que tiene por el inmediatamente superior o inferior (esta última decisión se toma aleatoriamente). En caso de que el gen a mutar fuese un parámetro  $S$ , se aplica la mutación binaria. Si el gen es de alguno de los componentes de codificación real (dominios y parámetros  $a$ ), el operador de mutación utilizado es el de Michalewicz [8].

## 5 Estudio experimental

Vamos a analizar el comportamiento de AG-Gr-P-OC en tres conjuntos de datos del repositorio UCI [2]: *Pima*, *Glass* y *Wisconsin breast cancer*, utilizados en otros trabajos de *one-class* [4, 13]. Para convertir esos tres problemas en problemas del paradigma *one-class*, se ha seleccionado siempre la clase menos representada dentro del conjunto de datos como clase anómala y el resto de clases se han unido en una única característica que es la clase objetivo. Se ha realizado una validación cruzada con 5 particiones distintas, y

se han utilizado las particiones disponibles dentro del repositorio KEEL [1], que están disponibles en la página web del proyecto KEEL (<http://www.keel.es/dataset.php>). Como dichas particiones estaban preparadas para clasificación multi-clase, el porcentaje de ejemplos en cada conjunto de entrenamiento no es exactamente del 80% ya que se han eliminado todos los ejemplos de la clase anómala y se han pasado a los respectivos conjuntos de test, que contienen más del 20% de ejemplos. La Tabla 1 tiene la información básica sobre los conjuntos de datos.

Tabla 1: Descripción resumida de los conjuntos de datos

Conj. de datos	#Ej.	#Var.	#Clases	Clase anómala
Glass1	214	9	7	6
Wisconsin	683	9	2	malignant
Pima	768	8	2	tested-positive

Con objeto de analizar el comportamiento de nuestra propuesta, comparamos los resultados obtenidos por AG-Gr-P-OC con los obtenidos por un método clásico del paradigma *one-class*, como son las máquinas vectores de soporte para *one-class* (OCSVM) [10] y con la versión previa del método propuesto aquí [14], que denominamos AG-Gr-OC. Los parámetros específicos del AG del método AG-Gr-P-OC se muestran a continuación, siendo  $N$  el número de variables del problema: Número de evaluaciones:  $500 \cdot N$ , Tamaño de población: 100, Probabilidad de cruce: 0.6, Probabilidad de mutación: 0.2, Parámetro *umbral\_cubrimiento*: 0.07, Parámetro *umbral\_soporte*: 0.07, Parámetro *umbral\_regla*: 0.25.

La Tabla 2 muestra los resultados obtenidos por los dos métodos (media de las cinco particiones que componen la validación cruzada). Para el caso del conjunto de entrenamiento se muestra solamente la tasa de acierto en clasificar la clase positiva mientras que para el conjunto de test se muestra la tasa de clasificación correcta total, las tasas para cada una de las dos clases (siendo  $CP$  la clase positiva y  $CA$  la clase anómala) y el AUC. Para el caso de los métodos que generan SBRDC se muestra también la media del número de reglas de los modelos obtenidos.

Como puede observarse, los dos métodos de generación de SBRDC obtienen mejores resultados que OCSVM en AUC sobre el conjunto de test, que es la medida más adecuada en este entorno ya que la detección correcta de la mayoría de las anomalías no debe derivar en bajos porcentajes de clasificación de ejemplos de la clase objetivo. Por otro lado, AG-Gr-P-OC obtiene unos resultados muy similares en AUC frente a AG-Gr-OC aunque reduce bastante el número de reglas, mejorando notablemente la interpretabilidad de los modelos obtenidos. Además, AG-Gr-P-OC obtiene mejor porcentaje de acierto que AG-Gr-OC sobre el total del conjunto de test, siendo algo peor el porcentaje de acierto sobre la clase positiva pero algo mejor el porcentaje de acierto sobre la clase anómala, por lo que funciona mejor como mecanismo de detección de anomalías, que es el en-

Tabla 2: Resultados detallados para los métodos analizados

Método	Conjunto de datos	Entrenamiento		Test			Número de reglas
		$Acierto_{CP}$	$Acierto_{CP}$	$Acierto_{CA}$	$Acierto_{Total}$	$AUC$	
OCSVM	Glass	0.0	0.0	1.0	0.171	0.5	–
	Wisconsin	0.583	0.598	1.0	0.891	0.799	–
	Pima	0.489	0.046	0.983	0.728	0.514	–
	Media	0.357	0.215	0.994	0.596	0.604	–
AG-Gr-OC	Glass	0.771	0.585	1.0	0.659	0.792	29.4
	Wisconsin	0.967	0.969	0.943	0.950	0.956	10.0
	Pima	0.718	0.696	0.625	0.645	0.661	31.4
	Media	0.818	0.750	0.856	0.751	0.803	23.6
AG-Gr-P-OC	Glass	0.667	0.619	0.911	0.672	0.765	7.2
	Wisconsin	0.954	0.941	0.994	0.979	0.968	7.0
	Pima	0.619	0.723	0.694	0.614	0.669	6.0
	Media	0.747	0.725	0.876	0.782	0.801	6.73

foque considerado para la distribución de la clase positiva y la clase anómala en los conjuntos de datos, tal y como se comentó en la sección 2.

## 6 Conclusiones

En este trabajo se ha presentado un sistema de Clasificación Basado en Reglas Difusas dentro del paradigma *one-class* utilizando un Algoritmo Genético para aprender una partición difusa adecuada para cada variable y un método específico para la generación de las reglas difusas que permite obtener modelos con buena precisión y muy interpretables. Como trabajo futuro, esperamos mejorar los resultados al ampliar el estudio a más conjuntos de datos.

## Agradecimientos

Este trabajo está soportado por el Ministerio de Ciencia e Innovación en el marco del proyecto TIN2011-28488 y por la Junta de Andalucía en el marco del proyecto P10-TIC-06858.

## Referencias

- [1] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera. KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. *Journal of Multi-Valued Logic and Soft Computing* 17(2-3):255–287, 2011.
- [2] A. Asuncion, D.J. Newman. UCI machine learning repository, 2007. University of California, Irvine, School of Information and Computer Sciences. <http://www.ics.uci.edu/mllearn/MLRepository.html>
- [3] J. E. Baker. Reducing bias and inefficiency in the selection algorithms. En: *Proc. of the Second International Conference on Genetic Algorithms (ICGA'87)*, pages 14–21, Hillsdale, 1987.
- [4] G. Cabral, A. Oliveira, C. Cahú. Combining nearest neighbor data description and structural risk minimization for one-class classification. *Neural Computation & Applications*, 18:175–183, 2009.
- [5] O. Cordón, F. Herrera, L. Magdalena, P. Villar. A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base. *Information Sciences*, 136(1-4):85–107, 2001.
- [6] F. Herrera, M. Lozano, J. Verdegay. Fuzzy connectives based crossover operators to model genetic algorithms population diversity. *Fuzzy Sets and Systems*, 92(1):21–30, 1997.
- [7] S. Khan, M. Madden. A Survey of Recent Trends in One Class Classification. En: *Lecture notes in Artificial Intelligence*, 6206:188–197, 2010.
- [8] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Program*. Springer-Verlag, 1996.
- [9] M. Moya, M. Koch, L. Hostetler. One-class classifier networks for target recognition applications. En: *Proc. of World Congress on Neural Networks*, pages 797–801, 1993.
- [10] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, R. Williamson. A Survey of Recent Trends in One Class Classification. En: *Neural Computation*, 13:1443–1471, 2001.
- [11] D. Tax, R. Duin. Uniform object generation for optimizing one-class classifiers. *Journal of Machine Learning Research* 2:155–173, 2001.
- [12] P. Thrift. Fuzzy logic synthesis with genetic algorithms. En: *Proc. Fourth International Conference on Genetic Algorithms (ICGA'91)*, 509–513, 1991.
- [13] J. Tian, H. Gu, C. Gao, J. Iian. Local density one-class support vector machines for anomaly detection. *Nonlinear Dynamics*, 64:127–139, 2011.
- [14] P. Villar, A. M. Sánchez, R. Montes, F. Herrera. Un Algoritmo Genético para aprendizaje de Sistemas de Clasificación Basados en Reglas Difusas dentro del paradigma *one-class*. *Actas CAEPIA'13*:1105–1114, 2013.