Un Modelo de Clasificación basado en Reglas de Asociación Difusas para Problemas de Alta Dimensionalidad y Selección Evolutiva de Reglas

Jesús Alcalá-Fdéz Rafael Alcalá Francisco Herrera

Dept. de Ciencias de la Computación e I.A., CITIC-UGR, Universidad de Granada, {jalcala,alcala,herrera}@decsai.ugr.es

Resumen

El aprendizaje de Sistemas de Clasificación Basados en Reglas Difusas en problemas de alta dimensionalidad se ve dificultado por el crecimiento exponencial que experimenta el espacio de búsqueda cuando el número de ejemplos y/o variables del problema es elevado. En este trabajo presentamos un método de clasificación basado en reglas de asociación difusas y selección evolutiva de reglas para problemas de alta dimensionalidad que nos permite obtener un clasificador preciso y compacto con un bajo coste computacional. Los resultados obtenidos sobre 26 problemas de distintas características muestran la efectividad del método propuesto.

Palabras Clave: Algoritmos Genéticos, Clasificación, Problemas de Alta Dimensionalidad, Reglas de Asociación Difusas.

1 INTRODUCCIÓN

Los Sistemas de Clasificación Basados en Reglas Difusas (SCBRDs) han sido empleados en muchas aplicaciones del mundo real, tales como detección de intrusiones [19], etc. Sin embargo, en muchas de estas áreas los datos de los que disponemos tienen un número de elevado de ejemplos y/o variables. Esto dificulta el aprendizaje de los SCBRDs debido al crecimiento exponencial que experimenta el espacio de búsqueda, dando lugar a problemas de escalabilidad y complejidad.

El descubrimiento de asociaciones es una de las técnicas de la minería de datos que más ha sido utilizada para extraer conocimiento interesante a partir de bases de datos grandes [10]. Durante los últimos años se han realizado muchos esfuerzos para hacer uso de sus ventajas en clasificación, dando lugar a un nuevo enfoque denominado clasificación asociativa [5, 7, 16, 20].

Un sistema de clasificación asociativa típico se construye en dos etapas:

- Extracción de las reglas de asociación a partir de la base de datos.
- Seleccionar un péqueño conjunto de reglas de asociación relevantes para construir el clasificador.

Con el objetivo de mejorar la interpretabilidad de las reglas obtenidas y eludir las fronteras no naturales en el particionamiento de los dominios de las variables, algunos investigadores han propuesto métodos para obtener clasificadores basados en reglas de asociación difusas [4, 12, 13].

En este trabajo nosotros presentamos un método de clasificación basado en reglas de asociación difusas y selección evolutiva de reglas para problemas de alta dimensionalidad, denominado FARC-HD, que permite obtener un clasificador preciso y compacto con un bajo coste computacional. Este método consta de 3 etapas:

1. Extracción de reglas de asociación difusas para clasificación.
2. Filtrado de las reglas candidatas.
3. Selección evolutiva de reglas.

Los resultados obtenidos de la comparación con un método reciente para obtener de un clasificador basado en reglas de asociación difusas, CFAR [4], sobre 26 problemas de distintas características muestran la efectividad del método propuesto. Además, este estudio ha sido contrastado a través de técnicas estadísticas no paramétricas.

La organización del trabajo es como sigue. En la siguiente sección se introduce brevemente las reglas de asociación difusas para clasificación. En la Sección 3 se presenta el método propuesto, describiendo en detalle cada una de sus etapas. La Sección 4 muestra el comportamiento del método propuesto en 26 problemas reales. Por último, en la Sección 5, se muestran algunas conclusiones.

2 PRELIMINARES: REGLAS DE ASOCIACIÓN DIFUSAS PARA CLASIFICACIÓN

Últimamente, la teoría de los conjuntos difusos ha sido utilizada más y más frecuentemente para describir relaciones entre los datos [14]. El uso de conjunto difusos nos permite extender los tipos de relaciones que se pueden representar [18], por lo que en los últimos años muchos investigadores han propuesto métodos para extraer reglas de asociación difusas a partir de datos cuantitativos [1, 6, 11].

Las medidas más comunes para medir el interés de una regla de asociación son el *soporte* y la *confianza*. Estas medidas pueden ser definidas para reglas de asociación difusas como:

$$Soporte(A \rightarrow B) = \frac{\sum_{x_p \in N} \mu_{AB}(x_p)}{\mid N \mid} \quad (1)$$

$$Confianza(A \rightarrow B) = \frac{\sum_{x_p \in N} \mu_{AB}(x_p)}{\sum_{x_p \in N} \mu_A(x_p)} \quad (2)$$

donde $\mid N \mid$ es el número total de ejemplos, $\mu_A(x_p)$ es el grado de emparejamiento del ejemplo x_p con el antecedente de la regla y $\mu_{AB}(x_p)$ es el grado de emparejamiento del ejemplo x_p con la regla completa.

Una regla de asociación difusa podría ser considerada una regla de clasificación difusa si el antecedente de la regla contiene los ítems difusos (cada ítem difuso es una tupla formada por una variable y un término lingüístico) y el consecuente una clase. Así, una regla de clasificación asociativa difusa como $A \rightarrow Clas_j$ podría ser medida en términos de soporte y confianza como:

$$Soporte(A \rightarrow Clas_j) = \frac{\sum_{x_p \in Clas_j} \mu_A(x_p)}{\mid N \mid} \quad (3)$$

$$Confianza(A \rightarrow Clas_j) = \frac{\sum_{x_p \in Clas_j} \mu_A(x_p)}{\sum_{x_p \in N} \mu_A(x_p)} \quad (4)$$

3 FARC-HD: UN CLASIFICADOR BASADO EN REGLAS DE ASOCIACIÓN DIFUSAS PARA PROBLEMAS DE ALTA DIMENSIONALIDAD

En esta sección se describe nuestra propuesta para obtener un clasificador basado en reglas de asociación difusas para problemas de alta dimensionalidad. Este método está basado en 3 etapas:

1. Extracción de reglas de asociación difusas para clasificación.
2. Filtrado de las reglas candidatas.
3. Selección evolutiva de reglas.

En las siguientes subsecciones introduciremos cada una de estas etapas, describiendo en detalle todas sus características.

3.1 EXTRACCIÓN DE REGLAS DE ASOCIACIÓN DIFUSAS PARA CLASIFICACIÓN

Para generar la base de reglas (BR) inicial empleamos un árbol de búsqueda para obtener los conjuntos de ítems difusos frecuentes para cada una de las clases. El nivel 0 de cada árbol estará formado por el conjunto vacío, mientras que el nivel 1 estará compuesto por todos los ítems difusos según un orden preestablecido. Cada nodo hijo de un nodo A será el resultado de unir dicho nodo con otro nodo del mismo nivel con el que se diferencia en un único ítem. La figura 1 muestra un ejemplo de un árbol obtenido a partir de dos variables (V_1 y V_2) con dos términos lingüísticos (B y A)

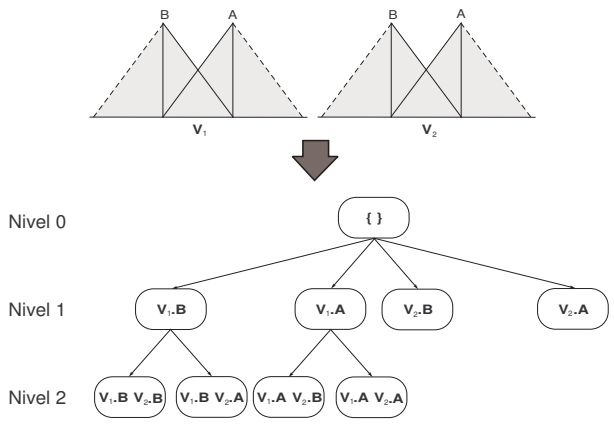

Figura 1: Árbol para las variables V_1 y V_2 con los términos lingüísticos B y A

Los conjuntos de ítems que presentan un soporte mayor que el soporte mínimo indicado por el experto son considerados frecuentes. Si el soporte del conjunto de ítems asociado a un nodo del árbol es menor que el soporte mínimo, este nodo no será extendido más debido a que ninguno de sus descendientes podrá tener un soporte mayor. De igual forma, si la confianza de un nodo es mayor que la confianza mínima indicada no será necesario extender más dicho nodo.

Una vez extraídos todos los conjuntos de ítems frecuentes generamos las reglas de clasificación utilizando los conjuntos de ítems frecuentes como antecedentes de las reglas y la clase como consecuente. Este proceso será repetido para cada una de las clases.

El número de conjuntos de ítems frecuentes depende directamente del soporte mínimo seleccionado. Este es calculado considerando el número total de ejemplos. Sin embargo, el número de ejemplos del que disponemos para cada una de las clases puede ser distinto. Por esta razón, el soporte mínimo es recalculado para cada una de las clases teniendo en cuenta el ratio de ejemplos disponible para cada clase:

$$minSop_{Clas_i} = minSop * f_{Clas_i} \tag{5}$$

En esta etapa el número de reglas generadas puede ser muy elevado. Sin embargo, es muy difícil para un ser humano manejar una gran cantidad de reglas. Además, si estas reglas tienen muchas variables en el antecedente son difíciles de interpretar. Por esta razón, la profundidad de los árboles será limita a un valor fijo ($Prof_{max}$) determinado por el experto.

3.2 FILTRADO DE LAS REGLAS CANDIDATAS

Para reducir el coste computacional, en esta etapa hacemos uso del descubrimiento de conjuntos para seleccionar las reglas más interesantes entre las reglas generadas en la etapa anterior por medio de un esquema de ejemplos ponderados [15]. Este esquema trata cada ejemplo de forma que cuando es cubierto por una regla seleccionada este no es eliminado. En su lugar el método almacena para cada ejemplo un contador i con el número de veces que ha sido cubierto por una regla seleccionada y decrementa su peso según la formula $p(e_j, i) = \frac{1}{i+1}$ (inicialmente el peso de cada ejemplo es 1,0). Un ejemplo será complemente eliminado cuando haya sido cubierto k_j veces.

Así, en cada iteración del método ordena las reglas según una medida de calidad y reduce el peso de los ejemplos cubiertos. Este proceso es repetido hasta que todos los ejemplos hayan sido cubiertos k_j veces o hasta que no queden más reglas para seleccionar.

En este trabajo nosotros hemos modificado la medida de calidad utilizada en [15] (wWRAcc') para poder manejar reglas difusas. Así, la medida de calidad para una regla $A \rightarrow Clas_j$ se define como:

$$\frac{n''(A \cdot Clas_j)}{n'(Clas_j)} \cdot \left(\frac{n''(A \cdot Clas_j)}{n''(A)} - \frac{n(Clas_j)}{N}\right) \tag{6}$$

donde $n''(A)$ representa el resultado de sumar el peso de cada ejemplo cubierto por su grado de emparejamiento con el antecedente de la regla, $n''(A \cdot Clas_j)$ el resultado de sumar el peso de cada ejemplo cubierto de la clase $Clas_j$ por su grado de emparejamiento con el antecedente de la regla y $n'(Clas_j)$ la suma de los pesos de los ejemplos de la clase $Clas_j$. Además, el primer término de la definición de wWRAcc' ha sido modificado por $\frac{n''(A \cdot Clas_j)}{n(Clas_j)}$ para premiar a aquellas reglas que cubran ejemplos no eliminados de la clase $Clas_j$.

3.3 SELECCIÓN EVOLUTIVA DE REGLAS

En esta etapa hacemos uso de un AG para seleccionar un subconjunto compacto de reglas con un elevada precisión de clasificación a partir de la BR generada en la etapa anterior. En este caso utilizamos el modelo genético del método CHC [8] que ha presentado buenos resultados en problemas de selección binaria [3]. A continuación se explicarán los componentes de este modelo: modelo evolutivo, codificación y población inicial, evaluación, operador de cruce y reinicialización.

3.3.1 Modelo Evolutivo

CHC hace uso de un enfoque de *Selección basado en Poblaciones*. P padres y sus hijos son combinados para seleccionar los P mejores individuos que formarán parte de la siguiente población. Este enfoque utiliza un mecanismo de prevención de incesto y de reinicialización para introducir diversidad en la población, en lugar del operador de mutación.

El mecanismo de prevención de incesto es utilizado para indicar cuando aplicar el operador de cruce, es decir, dos padres son cruzados sólo si su distancia de Hamming dividida por 2 es superior a un determinado umbral, L. Este valor umbral es inicializado como:

$$L = (\#Genes)/4.0 \tag{7}$$

donde $\#Genes$ es el número de genes que tiene un cromosoma. Cuando en una generación no entra ningún

individuo nuevo en la población, L es decrementado. En nuestro caso, L será decrementado cada vez un $\varphi\%$ de su valor inicial, siendo $\varphi\%$ determinado por el usuario (normalmente un 10%).

3.3.2 Codificación y población inicial

Cada cromosoma es un vector binario que determina cuando una regla es seleccionada o no ('1' y '0' respectivamente). Así, un cromosoma C tendrá la siguiente estructura:

$$Si \;\; c_i = 1 \;\; Entonces \;\; (R_i \in BR) \;\; sino \;\; (R_i \notin BR),$$

donde R_i es la i-ésima regla de la BR inicial. Para utilizar la información disponible, todas las reglas candidatas son incluidas en la población como una solución inicial. La población inicial se obtiene generando un individuo con todos los genes a '1', y el resto de los individuos son generados aleatoriamente dentro del conjunto {0, 1}.

3.3.3 Evaluación

Para evaluar un cromosoma calculamos el porcentaje de clasificación y minimizamos la siguiente función:

$$Funcion_{eval}(C) = clas\_por - w_1 \cdot NR \qquad (8)$$

donde NR es el número de reglas seleccionadas (para penalizar subconjuntos de reglas grandes) y w_1 es calculado utilizando el porcentaje de clasificación considerando todas las reglas de la BR inicial,

$$w_1 = \alpha \cdot (clas\_por_{inicial}/NR_{inicial}) \qquad (9)$$

donde α es un peso proporcionado por el experto que determina el equilibrio entre complejidad y precisión (una buena elección es 1,0). Un cromosoma será valorado con un valor $-w_1$ si hay alguna clase para la que no se ha seleccionado al menos una regla.

3.3.4 Operador de cruce

Nosotros utilizamos el operador de cruce HUX [9]. Este operador intercambia exactamente la mitad de los genes que son distintos en los padres, asegurando la máxima distancia entre los hijos y sus padres (exploración).

3.3.5 Reinicialización

Para salir de óptimos locales, CHC utiliza un mecanismo de reinicialización. En este caso, el mejor cromosoma se mantiene en la población y el resto son generados aleatoriamente dentro del conjunto {0, 1}. Este mecanismo será aplicado cuando el valor umbral L alcance un valor menor que 0.

4 EXPERIMENTOS Y ANÁLISIS DE RESULTADOS

Para analizar el funcionamiento del método propuesto hemos seleccionado 26 problemas reales con distintos tamaños. La tabla 1 resume las principales características de los 26 problemas y muestra el enlace al proyecto KEEL [2] desde el que podemos descargarlos. Para cada problema se muestra el número de Variables (Var), el número de ejemplos (Eje) y el número de clases (Cla).

Tabla 1: Problemas considerados en el análisis

Nombre	Var	Eje	Cla	Nombre	Var	Eje	Cla
Iris	4	150	3	Crx	15	690	2
Phoneme	5	5404	2	Pen-based	16	10992	10
Monks	6	432	2	Segment	19	2310	7
Ecoli	7	336	8	German	20	1000	2
Pima	8	768	2	Twonorm	20	7400	2
Yeast	8	1484	10	Ringnorm	20	7400	2
Glass	9	214	7	Thyroid	21	7200	3
Contracep.	9	1473	3	Wdbc	30	569	2
Page-blocks	10	5472	5	Ionos.	34	351	2
Magic	10	19020	2	Dermato.	34	366	6
Wine	13	178	3	SatImage	36	6435	6
Heart	13	270	2	Spambase	57	4597	2
Cleveland	13	303	5	Sonar	60	208	2

Disponibles en `http://sci2s.ugr.es/keel/datasets.php`

En este estudio nosotros comparamos el método propuesto con el método CFAR [4]. El método está basado en 2 etapas. En la primera genera todas las reglas de asociación difusas posibles mediante una extensión del método Apriori y, en la segunda selecciona las mejores reglas para construir el clasificador.

En los distintos experimentos, se trabaja con un modelo de validación cruzada con 10 particiones de datos, esto es, 10 particiones aleatorias al 10%, y la combinación de cuatro particiones (90%) como entrenamiento y una partición como test. Así se tienen 10 particiones al 90% y 10% en entrenamiento y test. Para cada partición cada método ha sido ejecutado 3 veces y en las tablas se muestran los resultados medios de las 30 ejecuciones de cada algoritmo. Además, hemos aplicado el test no paramétrico de Rangos y Signos de Wilcoxon [17] con un nivel de confianza del 95% ($\alpha = 0,05$) para asegurar que existen diferencias significativas entre los métodos.

Las particiones lingüísticas iniciales consideradas tendrán cinco términos lingüísticos con forma triangular. Los valores de los parámetros utilizados en todos los experimentos son:

Tabla 2: Resultados obtenidos por los métodos CFAR y FARC-HD

	CFAR				FARC-HD			
Nombre	**#R**	**#L**	**Ent**	**Test**	**#R**	**#L**	**Ent**	**Tst**
Iris	9,1	1,5	91,3	90,7	4,0	1,0	97,4	**96,4**
Phoneme	2,0	1,0	70,7	70,7	6,2	1,7	80,2	**80,1**
Monks	12,1	1,8	100,0	99,6	4,9	1,2	99,7	**99,8**
Ecoli	4,0	3,1	48,5	47,6	26,0	2,4	86,1	**78,0**
Pima	2,0	1,9	65,1	65,1	6,6	2,1	77,0	**75,9**
Yeast	2,0	1,0	16,4	16,4	20,8	2,6	59,0	**55,5**
Glass	3,4	2,1	44,2	44,7	21,0	2,5	77,5	**64,0**
Contracep.	2,0	1,0	42,7	42,7	22,8	2,6	58,1	**52,0**
Page-blocks	2,0	1,0	89,8	89,8	7,9	2,0	94,0	**93,8**
Magic	2,0	1,5	64,8	64,8	8,9	1,9	80,4	**80,4**
Wine	115,7	3,0	99,6	**93,2**	10,2	1,8	98,9	92,1
Heart	45,3	3,3	86,0	82,2	17,4	2,5	92,0	**83,2**
Cleveland	2,0	2,1	53,9	53,9	39,0	2,9	81,8	**56,5**
Crx	89,9	5,1	89,5	**86,8**	9,8	2,3	89,4	85,5
Pen-based	6,9	2,9	36,5	36,4	53,6	2,7	94,2	**93,6**
Segment	43,8	5,1	56,9	56,9	22,2	2,4	91,1	**89,9**
German	2,0	2,7	70,0	70,0	23,0	2,5	80,5	**71,4**
Twonorm	1981,7	3,7	92,0	**91,7**	11,9	2,1	88,4	87,5
Ringnorm	-	-	-	-	9,8	1,5	88,2	87,9
Thyroid	67,6	5,8	92,6	92,6	3,7	2,5	93,6	**93,4**
Wdbc	-	-	-	-	6,3	1,2	95,4	**94,4**
Ionos.	-	-	-	-	15,2	1,9	95,4	**90,0**
Dermato.	-	-	-	-	23,5	2,4	99,3	**90,2**
SatImage	-	-	-	-	19,0	2,7	75,6	**74,9**
Spambase	-	-	-	-	9,8	1,7	84,8	**84,8**
Sonar	37,3	3,0	83,0	72,5	11,6	2,4	93,8	**78,9**
Media	121,6	2,6	69,7	68,4	16,0	2,1	86,6	**81,9**

- FARC-HD: $MinSop = 0,05$, $MinConf = 0,8$, $Prof_{max} = 3$, $k_t = 2$, $Pob = 50$, $Eval = 5.000$, $\alpha = 0,02$
- CFAR: $Minpsop = \{0,05, 0,1\}$, $Minpconf = 0,85$, $MS = 0,15$

Destacar que los autores del método CFAR utilizaron $0,1$ como soporte mínimo, lo que podría ser muy elevado para algunos de los problemas utilizados en este estudio. Para evitar este problema este método ha sido ejecutado utilizando dos soportes mínimos ($0,05$ y $0,1$), mostrando en las tablas el mejor resultado obtenido con cada uno de ellos en cada problema.

Los resultados obtenidos se muestran en la Tabla 2, donde $\#R$ representa el número medio de reglas, $\#L$ el número medio de términos lingüísticos en el antecedente de las reglas y, Ent y $Test$ el porcentaje medio de acierto sobre las particiones de entrenamiento y test. Hay que destacar que el método CFAR no puede ejecutarse en 6 de los 26 problemas utilizados en el estudio.

Como podemos observar, el método propuesto obtiene la mejor precisión en media. Además, el número medio de reglas obtenidas es reducido y con menos de 3 términos lingüísticos en el antecedente, lo que facilita su interpretabilidad desde el punto de vista del usuario.

La tabla 3 muestra los resultados de aplicar el test de Wilcoxon sobre los resultados obtenidos en test. Como podemos observar la hipótesis nula es rechazada ($valor\text{-}p < \alpha$) y nuestro método es el que obtiene mayor rango, por lo que podemos concluir que nuestro método es el que presenta un mejor funcionamiento.

Tabla 3: Test de Wilcoxon. Rango positivo para FARC-HD (R^+). Rango negativo para CFAR (R^-).

Comparación	R^+	R^-	**Hipótesis** ($\alpha = 0,05$)	**valor-p**
FARC-HD vs. CFAR	192	18	**Rechazada**	0,001

En la tabla 4 se muestran los tiempos de ejecución obtenidos en un Pentium Corel 2 Quad a 2.5GHz, con 4Gb de memoria y con sistema operativo linux. Si analizamos estos tiempos podemos ver como el método CFAR emplea una gran cantidad de tiempo cuando se incrementa el tamaño del problema, mientras que el método propuesto presenta un bajo coste computacional en todos los problemas.

Tabla 4: Tiempos de ejecución (hh:mm:ss)

Nombre	CFAR	FARC-HD	Nombre	CFAR	FARC-HD
Iris	00:00:00	00:00:00	Crx	00:02:02	00:00:02
Phoneme	00:00:05	00:00:13	Pen-based	00:00:07	00:03:15
Monks	00:00:01	00:00:01	Segment	03:08:57	00:00:26
Ecoli	00:00:01	00:00:01	German	00:02:17	00:00:06
Pima	00:00:06	00:00:02	Twonorm	05:03:33	00:01:06
Yeast	00:00:01	00:00:08	Ringnorm	-	00:01:06
Glass	00:00:04	00:00:01	Thyroid	08:57:26	00:00:13
Contracep.	00:00:01	00:00:05	Wdbc	-	00:00:09
Page-blocks	00:01:08	00:00:15	Ionos.	-	00:00:04
Magic	00:16:30	00:01:00	Dermato.	-	00:00:04
Wine	00:00:45	00:00:01	SatImage	-	00:04:51
Heart	00:00:12	00:00:01	Spambase	-	00:03:29
Cleveland	00:00:09	00:00:02	Sonar	00:00:20	00:00:36

5 CONCLUSIONES

En este trabajo hemos presentado un método de clasificación basado en reglas de asociación difusas y selección evolutiva de reglas para problemas de alta di-

mensionalidad, FARC-HD, que nos permite obtener un clasificador preciso y compacto con bajo coste computacional. Si observamos los resultados obtenidos podemos ver como el método propuesto obtiene en media los mejores resultados en precisión y presenta un bajo coste computacional en todos los problemas, obteniendo una buena escalabilidad. Además, este método obtiene modelos con un número reducido de reglas y que incluyen pocas variables en el antecedente, por lo que facilita su interpretabilidad desde el punto de vista de usuario.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio Español de Educación y Ciencia bajo el proyecto TIN2008-06681-C06-01.

Referencias

[1] J. Alcalá-Fdez, R. Alcalá, M. Gacto, F. Herrera. Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms. *Fuzzy Sets and Systems* 160:7, Pág.905-921, 2009.

[2] J. Alcalá-Fdez, L. Sánchez, S. García, M. del Jesus, S. Ventura, J. Garrell, J. Otero, C. Romero, J. Bacardit, V. Rivas, J. Fernández, F. Herrera. KEEL: A software tool to assess evolutionary algorithms to data mining problems. *Soft Computing* 13:3, Pág.307-318, 2009.

[3] J. Cano, F. Herrera, M. Lozano. Using evolutionary algorithms as instance selection for data reduction in kdd: an experimental study. *IEEE Transactions on Evolutionary Computation* 7:6, Pág.561-575, 2003.

[4] Z. Chen, G. Chen. Building an associative classifier based on fuzzy association rules. *International Journal of Computational Intelligence Systems* 1:3, Pág.262-273, 2008.

[5] G. Chen, H. Liu, L. Yu, Q. Wei, X. Zhang. A new approach to classification based on association rule mining. *Decision Support Systems* 42:2, Pág.674-689, 2006.

[6] M. Delgado, N. Marín, D. Sánchez, M. Vila. Fuzzy association rules: General model and applications. *IEEE Transactions on Fuzzy Systems* 11:2, Pág.214-225, 2003.

[7] G. Dong, X. Zhang, L. Wong, J. Li. Caep: Classification by aggregating emerging patterns. *Proceedings of the Second International Conference on Discovery Science (DS)*, Lecture Notes in Artificial Intelligence 1721, Pág.30-42, 1999.

[8] L.J. Eshelman. The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. *Foundations of genetic Algorithms* 1, Pág.265-283, 1991.

[9] L.J. Eshelman, J.D. Schaffer. Real-coded genetic algorithms and interval-schemata. *Foundations of Genetic Algorithms* 2, Pág.187-202, 1993.

[10] J. Han, M. Kamber. Data Mining: Concepts and Techniques. Second Edition. *Morgan Kaufmann*, 2006.

[11] T. Hong, Y. Lee. An overview of mining fuzzy association rules. *Studies in Fuzziness and Soft Computing* 220, Pág.397-410, 2008.

[12] Y.-C. Hua, G.-H. Tzeng. Elicitation of classification rules by fuzzy data mining. *Engineering Applications of Artificial Intelligence* 16:7-8, Pág.709–716, 2003.

[13] Y.-C. Hua. Determining membership functions and minimum fuzzy support in finding fuzzy association rules for classification problems. *Knowledge-Based Systems* 19:1, Pág.57-66, 2006.

[14] H. Ishibuchi, T. Nakashima, M. Nii. Classification and Modeling with Linguistic Information Granules. *Advanced Approaches to Linguistic Data Mining, Springer-Verlag*, 2004.

[15] B. Kavsek, N. Lavrac. Apriori-sd: Adapting association rule learning to subgroup discovery. *Applied Artificial Intelligence* 20:7, Pág.543-583, 2006.

[16] B. Liu, Y. Ma, C. Wong. Classification using association rules: weaknesses and enhancements. *Data Mining for Scientific and Engineering Applications* 2, Pág.591-602, 2001.

[17] D. Sheskin. Handbook of parametric and nonparametric statistical procedures. *Chapman & Hall/CRC*, 2003.

[18] T. Sudkamp. Examples, counterexamples, and measuring fuzzy associations. *Fuzzy Sets and Systems* 149:1, Pág.57-71, 2005.

[19] C. Tsang, S. Kwong, H. Wang. Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection. *Pattern Recognition* 40:9, Pág.2373-2391, 2007.

[20] B. Liu, W. Hsu, Y. Ma. Integrating classification and association rule mining. *Proceedings of the International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, Pág.80-86, 1998.