Aprendizaje Supervisado de Reglas Difusas mediante un Sistema Clasificador Evolutivo Estilo Michigan

Albert Orriols-Puig
Grup de Recerca en Sistemes Intel·ligents
Universitat Ramon Llull
08022, Barcelona
aorriols@salle.url.edu

Jorge Casillas
Dpto. de Ciencias de la Computación e IA
Universidad de Granada
18071, Granada
casillas@decsai.ugr.es

Ester Bernadó-Mansilla
Grup de Recerca en Sistemes Intel·ligents
Universitat Ramon Llull
08022, Barcelona
esterb@salle.url.edu

Resumen

Este artículo presenta Fuzzy-UCS, un sistema clasificador de la familia de Michigan diseñado para tratar problemas de aprendizaje supervisado. Fuzzy-UCS se inspira en el sistema clasificador UCS para crear una arquitectura de aprendizaje incremental de reglas difusas. En este trabajo se presenta detalladamente el sistema, se prueba sobre un conjunto de problemas de clasificación y se compara con otros sistemas de aprendizaje. Los resultados muestran que Fuzzy-UCS tiene un rendimiento equiparable al de UCS y otras técnicas de aprendizaje automático ampliamente utilizadas, y que la representación difusa permite una mayor interpretabilidad de las reglas evolucionadas. Estos resultados prometedores abren líneas de investigación futura en el uso de sistemas clasificadores Michigan con representación difusa en aprendizaje supervisado.

1. Introducción

Los sistemas clasificadores de la familia de Michigan (LCS) [7, 8], son máquinas de aprendizaje que usan *algoritmos genéticos* [7, 6] para evolucionar un conjunto de reglas de producción que colaboran para predecir la clase de nuevos ejemplos. Una de las características más destacadas de los LCS es que evolucionan el conocimiento incrementalmente; partiendo de un conjunto de reglas de producción inicializada de manera aleatoria, los LCS usan un algoritmo evolutivo para crear reglas más precisas y generales.

Típicamente, los LCS de la familia Michigan han usado representaciones basadas en reglas intervalares para tratar con datos continuos, y la investigación entorno a ellos se ha centrado en aumentar su precisión. Esto ha resultado en sistemas altamente competitivos en la disciplina del aprendizaje automático, pero su utilización real se ha visto mermada por la pobre interpretabilidad de los conjuntos de reglas evolucionados. Para aliviar este problema, recientemente, la comunidad de LCS ha empezado a prestar atención en los sistemas de reglas difusos, los cuales proveen una metodología robusta para tratar con datos imprecisos e incompletos, garantizando una alta interpretabilidad del conocimiento evolucionado. La convergencia de ambas áreas ha resultado en los primeros sistemas clasificadores Michigan que usan una representación basada en reglas difusas, como por ejemplo los sistemas descritos en [16, 4]. La mayoría de estos sistemas se han diseñado para solucionar problemas de control y aprendizaje por refuerzo.

En este artículo se afronta el problema de la interpretabilidad en LCS y se propone Fuzzy-UCS, un sistema clasificador con representación difusa que funciona bajo una arquitectura de aprendizaje supervisado. Con esta finalidad, se parte del sistema clasificador UCS [2], el cual ha demostrado ser altamente competi-

tivo respecto algunas de las técnicas más usadas en aprendizaje supervisado como el árbol de decisión C4.5 [13] y la máquina de soporte vectorial SMO [12]. Con la inclusión de reglas difusas en Fuzzy-UCS, se persigue una mejor interpretabilidad del conocimiento evolucionado mientras se mantiene un rendimiento similar al de UCS.

El resto del documento se estructura del siguiente modo. La sección 2 describe detalladamente el sistema Fuzzy-UCS. En la sección 3 se prueba el sistema Fuzzy-UCS sobre un conjunto de problemas del mundo real, y se compara su precisión con varios sistemas de aprendizaje. Finalmente, la sección 4 resume el trabajo, concluye y abre líneas de investigación futuras.

2. Descripción de Fuzzy-UCS

La figura 1 muestra esquemáticamente el funcionamiento de Fuzzy-UCS. El sistema trabaja en dos modos distintos: aprendizaje o exploración y prueba o explotación. En la fase de aprendizaje, Fuzzy-UCS evoluciona un conjunto de reglas difusas precisas y máximamente generales. En fase de prueba, se usa el conocimiento evolucionado para inferir la clase de nuevos ejemplos.

2.1. Representación del Conocimiento

El sistema clasificador Fuzzy-UCS evoluciona una población de *clasificadores* o individuos, los cuales están formados por una regla difusa con antecedente en forma normal conjuntiva y una serie de parámetros. La regla difusa se representa del siguiente modo:

SI x_1 es $\widetilde{A_1^k}$ y $\cdots$ y x_n es $\widetilde{A_n^k}$ **ENTONCES** c_1 **CON** w_1^k, $\cdots$, c_m **CON** w_m^k

Cada variable x_i se representa con una disyunción de etiquetas lingüísticas $\widetilde{A_i^k} = \{A_{i1} \vee \cdots \vee A_{in_i}\}$, y el consecuente mantiene, por cada clase j, un peso w_j^k que indica el grado de certeza con el que la regla k predice la clase j. Cabe destacar que este tipo de regla intrínsecamente permite generalización ya que cada variable puede tener un número arbitrario de etiquetas lingüísticas. Así pues, se puede simular la ausencia de una variable x_i de una regla k si $\widetilde{A_i^k}$ tiene todas las etiquetas lingüísticas.

Cada clasificador tiene cuatro parámetros básicos: a) el *fitness* F, que estima la precisión de la regla, b) el tamaño del conjunto de reglas correctas cs, que promedia el número de clasificadores que hay en los conjuntos de reglas correctas en los que la regla participa, c) la experiencia exp, que recuenta las contribuciones de la regla para clasificar los ejemplos con los que se empareja y d) la numerosidad, que indica el número de copias de la regla en la población.

El cálculo del *grado de emparejamiento* $\mu_{A^k}(e)$ de una regla k con un ejemplo e es como sigue. Por cada variable x_i se computa el grado de pertenencia para cada una de sus etiquetas lingüísticas, y éstos valores se agregan mediante una T-conorma (disyunción). El grado de pertenencia $\mu_{A^k}(e)$ resulta de aplicar una T-norma (conjunción) sobre todos las variables de la regla. En los experimentos realizados, se ha usado la suma acotada ($min\{1, a+b\}$) como T-conorma y el producto ($a \cdot b$) como T-norma.

Para representar la condición de una regla, cada variable se codifica con una cadena binaria de longitud ℓ (siendo ℓ el número de etiquetas lingüísticas máximas que puede tomar la variable). Cuando un bit vale 1, se usa la etiqueta lingüística correspondiente. Los pesos w_j^k del consecuente se codifican en un vector de números reales. Por ejemplo, si se tienen las tres etiquetas lingüísticas {P (pequeño), M (mediano), G (grande)} por variable y dos clases posibles, la regla

SI x_1 es P y x_2 es {P o G} **ENTONCES** c_1 **CON** 0,8 , c_2 **CON** 0,2

se codifica como: [100|101||0,8|0,2].

2.2. Fase de Aprendizaje

La fase de aprendizaje de Fuzzy-UCS se ha adaptado de UCS [2] teniendo en cuenta las diferencias que supone tratar con reglas difusas. Las dos diferencias más importantes se encuentran en a) la función de emparejamiento

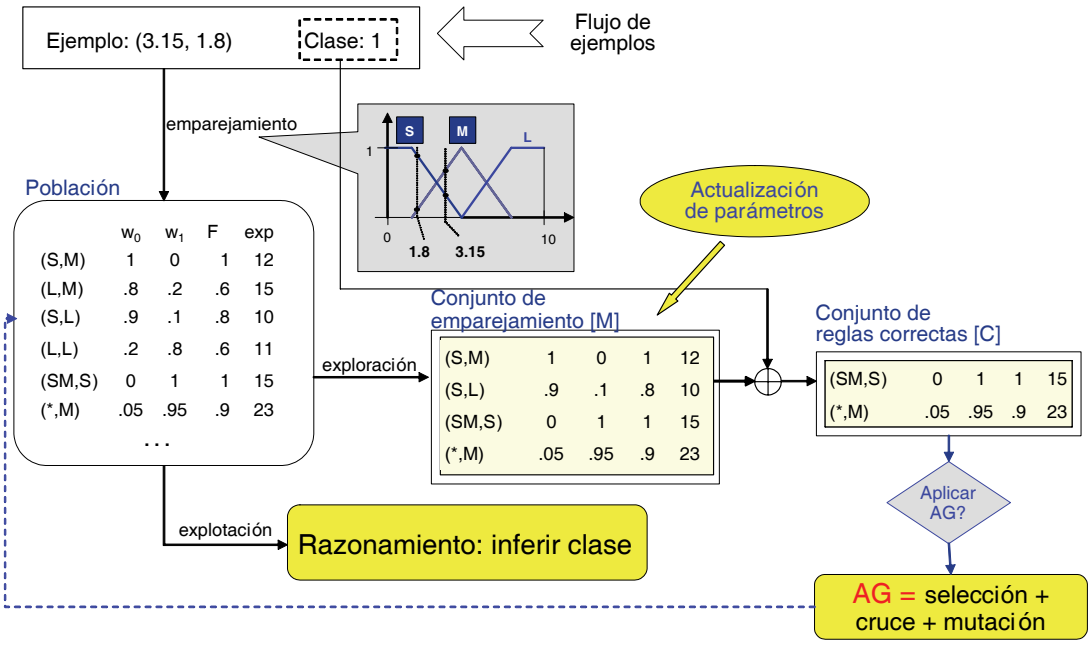

Figura 1: Ilustración esquemática de Fuzzy-UCS. El ciclo de la ejecución depende de si el sistema está en fase de aprendizaje (exploración) o prueba (explotación).

de una regla con un ejemplo y b) la inferencia de la clase para un ejemplo desconocido. En primer lugar, mientras que en UCS la función de emparejamiento indica si la regla se empareja o no con un ejemplo concreto, en Fuzzy-UCS ésta retorna el grado de emparejamiento $\mu_{A^k}(e)$ de la regla k con el ejemplo e (donde $0 \leq \mu_{A^k}(e) \leq 1$). En segundo lugar, dado un conjunto final de reglas difusas, Fuzzy-UCS debe combinar el conocimiento de las reglas para predecir la clase de un nuevo ejemplo.

Fuzzy-UCS aprende incrementalmente como se explica seguidamente. En cada iteración de aprendizaje, Fuzzy-UCS recibe un ejemplo e con su clase asociada c y forma el *conjunto de emparejamiento* [M], que contiene todas las reglas que tienen un grado de emparejamiento con el ejemplo mayor que cero. Seguidamente, se crea el *conjunto de reglas correctas* [C], donde se insertan todas las reglas de [M] que su peso máximo w_c^k corresponda al de la clase c.

Si no hay ningún clasificador en [C] que se empareje con e con grado máximo, se dispara el *operador de cubrimiento*. El operador de cubrimiento crea la regla con máximo grado de emparejamiento con e mediante el procedimiento siguiente. Se crea una regla nueva y, por cada variable de ésta, se agrega la etiqueta lingüística que maximiza el emparejamiento con el valor de entrada correspondiente a la variable. Además, para permitir una mayor generalización en las reglas iniciales, se decide si se introduce cada una del resto de etiquetas lingüísticas con probabilidad $P\#$. La regla resultante se inserta en la población.

2.3. Actualización de Parámetros

Al final de cada iteración de aprendizaje se actualizan los parámetros de las reglas en [M]. Primeramente, se computa la suma de empa-

rejamientos cm_j de cada regla k por cada clase j:

$$cm_j^k = cm_j^k + m(k,j) \quad (1)$$

donde

$$m(k,j) = \begin{cases} \mu_{A^k}(e) & \text{si } j = c \\ 0 & \text{e.o.c} \end{cases} \quad (2)$$

Seguidamente, se calcula la suma de todos los grados de emparejamiento sm:

$$sm^k = sm^k + \mu_{A^k}(e) \quad (3)$$

Y luego, se computa el peso w_j^k por cada clase j:

$$w_j^k = \frac{cm_j^k}{sm^k} \quad (4)$$

Claramente, la suma de todos los pesos w_j^k es 1. Por ejemplo, si la regla k sólo se empareja con ejemplos de la clase j, independientemente del grado de emparejamiento con cada ejemplo, el peso w_j^k valdrá 1 mientras que el resto de pesos serán 0. Reglas que se emparejen con ejemplos de distintas clases tendrán pesos con valores comprendidos entre 0 y 1.

El *fitness* de una regla se computa a partir de los pesos por clase con la finalidad de favorecer las reglas que se emparejan mayoritariamente con ejemplos de una sola clase [9]:

$$F^k = w_{max}^k - \sum_{j|j \neq max} w_j^k \quad (5)$$

La ecuación substrae del peso con valor máximo la suma del resto de pesos. Cabe destacar que esta fórmula permite clasificadores con *fitness* negativo o zero. Por ejemplo, una regla que tenga pesos con el mismo valor y más de dos clases resultará en un *fitness* negativo. Como se verá más adelante, estas reglas no serán tenidas en cuenta ni en el algoritmo genético ni en el proceso de inferencia de una nueva clase. Finalmente, la experiencia de cada regla se calcula como la suma de los grados de emparejamiento de las reglas en cada participación a [M].

2.4. Descubrimiento de nuevas Reglas

Fuzzy-UCS usa un algoritmo genético (AG) para descrubir nuevas reglas prometedoras. El AG se aplica en [C], proporcionando de esta manera un sistema de *niching* intrínseco. El AG se dispara si el tiempo medio desde la aplicación del último AG sobre las reglas en [C] es mayor que el umbral θ_{GA}. En este caso, el AG selecciona dos reglas de [C] con probabilidad proporcional al producto de una potencia de su *fitness* y el grado de emparejamiento con el ejemplo a partir del cual se ha creado [C]: $\mu_{A^k}(e) \cdot F^\nu$, donde ν es un parámetro que permite variar la presión hacia reglas con más *fitness*. El *fitness* de reglas poco experimentadas se disminuye hasta que sus parámetros no se actualicen un número mínimo de veces.

Los dos padres seleccionados se cruzan y mutan con probabilidad χ y μ respectivamente, generando dos nuevos hijos. El operador de cruce cruza los dos padres por dos puntos de corte. El operador de mutación decide para cada variable si se debe mutar. En este caso, selecciona aleatoriamente entre tres tipos de mutación: expansión, contracción o desplazamiento. La expansión añade una etiqueta lingüística a la variable; por lo tanto, sólo se puede aplicar en variables que no tengan todas las etiquetas lingüísticas. La contracción es el proceso contrario: elimina una etiqueta lingüística de una variable; así pues, sólo es aplicable sobre variables que tengan más de una etiqueta lingüística. El desplazamiento elimina una etiqueta lingüística de una variable y añade su vecino.

Finalmente, los dos hijos se introducen en la población. Para cada hijo, se busca en [C] si hay alguna regla que lo pueda subsumir; es decir, una regla k suficientemente experimentada ($exp_k > \theta_{sub}$) y precisa ($F^k > F_0$) las variables de cuya condición tengan, como mínimo, las mismas etiquetas lingüísticas que las del hijo. En este caso, se incrementa la numerosidad de la regla subsumidora. Contrariamente, el hijo se inserta en la población, eliminando una regla si la población está llena. Cada regla tiene una probabilidad de ser eliminada proporcional a:

$$P_{del_k} = \begin{cases} cs_k \cdot num_k \cdot \frac{\overline{F}}{F_k} & \text{si } exp > \theta_{del} \text{ y } \delta\overline{F} > F_k \\ cs_k \cdot num_k & \text{e.o.c} \end{cases} \quad (6)$$

donde $\overline{F}$ es el *fitness* medio de las reglas en [P], y δ y θ_{del} son dos parámetros de configuración ($0 < \delta < 1$ y $\theta_{del} > 0$). Es decir, el método presiona hacia la eliminación de reglas numerosas que pertenezcan a conjuntos grandes de reglas correctas; además, penaliza las reglas experimentadas con *fitness* bajo.

2.5. Mecanismo de Razonamiento

En modo de explotación o prueba, dado un nuevo ejemplo de entrada e con clase desconocida, Fuzzy-UCS debe combinar el conocimiento de las reglas difusas evolucionadas para predecir la clase del ejemplo. En la literatura se pueden encontrar distintas propuestas de métodos de razonamiento, y se ha mostrado que, en general, los métodos que combinan la información contenida en un conjunto de reglas resultan en rendimientos más altos que las que escogen la regla con grado de emparejamiento y *fitness* más alto [5]. Por este motivo, en Fuzzy-UCS se ha diseñado un proceso que infiere la clase a partir de los votos de todas las reglas suficientemente experimentadas ($exp > \theta_{exploit}$) y con *fitness* positivo que tienen un grado de emparejamiento mayor que zero con el ejemplo de entrada. Específicamente, cada regla vota por cada clase j proporcionalmente al grado de emparejamiento y al peso por clase: $\mu_{A^k}(e) \cdot w_j^k$.

3. Resultados Experimentales

Esta sección evalúa el comportamiento de Fuzzy-UCS respecto al de otras técnicas de aprendizaje automático.

3.1. Metodología de Experimentación

Se ha seleccionado un conjunto de 10 problemas extraídos del repositorio UCI [3], cuyas características se resumen en la tabla 1. Se ha usado el método de validación *10-fold cross-validation* [15] para dividir el conjunto inicial de datos en diez conjuntos de entrenamiento y sus correspondientes conjuntos de prueba. Además, se ha usado el porcentaje de aciertos en prueba como métrica para evaluar el rendimiento de los algoritmos de aprendizaje.

Cuadro 1: Propiedades de los conjuntos de entrenamiento. Cada columna describe respectivamente: el nombre del problema (prob), el número de ejemplos (#ej), en número de atributos (#at), el número de clases (#c) y el porcentaje de ejemplos con valores desconocidos (%dsc).

prob	**#ej**	**#at**	**#c**	**%dsc**
annealing	898	38	5	0,9%
balance	625	4	3	7,8%
bupa	345	6	2	42,0%
glass	214	9	6	4,2%
heart-c	303	13	2	45,5%
heart-s	270	13	2	44,4%
iris	150	4	3	33,3%
wbcd	699	9	2	34,5%
wine	178	13	3	27%
zoo	101	17	7	4,0%

Se ha comparado Fuzzy-UCS con los algoritmos de aprendizaje con representación difusa Fuzzy GP y Fuzzy LogitBoost y no difusa C4.5, SMO y UCS. Fuzzy GP [14] es un algoritmo de programación genética que construye un sistema clasificador difuso. Fuzzy LogitBoost [11] es un algoritmo de *boosting* que iterativamente invoca un algoritmo genético que evoluciona reglas difusas débiles. C4.5 [13] es un árbol de decisión que aumenta las capacidades del ID3 pues permite manejar datos continuos y desconocidos. SMO [12] es una implementación de las máquinas de soporte vectorial [17]; en nuestros experimentos usamos un *kernel* lineal. UCS es el sistema clasificador en el cual se inspira Fuzzy-UCS. Para ejecutar Fuzzy-GP y Fuzzy-GAP se utilizó la implementación disponible en la plataforma KEEL [1]. Se usó la configuración por defecto, pero fijando el número de etiquetas lingüísticas a 5. Para ejecutar C4.5 y SMO se usó Weka [18]. Ambos métodos se configuraron por defecto. UCS se eje-

cutó con un software propio y con la siguiente configuración (ver [2, 10] para detalles de notación): acc_0=0,99, ν=10, θ_{GA}=50, χ=0,8, μ=0,04, θ_{del}=50, θ_{sub}=50, δ=0,1 y $P_{\#}$=0,6. En Fuzzy-UCS se usó una configuración parecida: F_0=0,99, ν=10, θ_{GA}=50, χ=0,8, μ=0,04, θ_{del}=50, θ_{sub}=50, δ=0,1 y $P_{\#}$=0,6. Igual que en el resto de sistemas de aprendizaje difusos, el número de etiquetas lingüísticas por variable se fijó a 5.

3.2. Resultados Obtenidos

La tabla 3.2 muestra los resultados obtenidos con los seis sistemas de aprendizaje. En los métodos estocásticos, es decir, Fuzzy GP, Fuzzy LogitBoost, UCS y Fuzzy-UCS, los resultados se corresponden a la media de 10 ejecuciones con semillas aleatorias distintas. Para analizar los resultados en global, la última columna de la tabla recuenta el *ranking* medio de cada algoritmo, calculado con el procedimiento que se explica a continuación. Por cada conjunto de entrenamiento se ordenan los algoritmos según sus resultados, de modo que se le da la primera posición del *ranking* al algoritmo con un porcentaje de acierto más alto, al segundo algoritmo se le asigna la segunda posición y así sucesivamente. En caso que haya un empate entre dos o más algoritmos, se les asigna la posición media a cada uno de ellos. El promedio de las posiciones en el *ranking* para cada problema indica la calidad de un algoritmo de aprendizaje respecto al resto de algoritmos de la comparativa.

SMO presenta el mejor comportamiento con un *ranking* medio de 2,3, seguido muy de cerca por UCS, con un *ranking* medio de 2,45. Fuzzy-UCS es el tercero con un *ranking* de 2,75. Ya más alejados, C4.5, Fuzzy LogitBoost y Fuzzy-GA ocupan la cuarta, quinta y sexta posición respectivamente. Nótese la competitividad de Fuzzy-UCS respecto a las otras técnicas de aprendizaje. Fuzzy-UCS ocupa una posición media muy cercana a la obtenida por el mejor algoritmo, SMO, y sólo es superado por técnicas que no usan representación difusa. Eso indica que Fuzzy-UCS es un sistema muy robusto, pues en media siempre esta en el grupo de los mejores algoritmos de la comparativa. Además, se observa que Fuzzy-UCS supera ampliamente a los dos métodos con representación difusa. Igualmente, también supera en global al árbol de decisión C4.5, uno de los algoritmos más utilizados en el campo del aprendizaje automático.

Los buenos resultados de SMO se ven deslucidos por la poca interpretabilidad del conocimiento creado. SMO representa el conocimiento en forma de pesos de las máquinas de soporte vectorial; por lo tanto, el experto humano puede sacar poco conocimiento. El mejor comportamiento de UCS respecto Fuzzy-UCS ya era esperado, pues el modo de aprendizaje de ambos algoritmos es similar, pero UCS representa el conocimiento mediante reglas intervalares. Consecuentemente, la representación de UCS tiene más libertad que la de Fuzzy-UCS, donde todas las variables comparten la misma semántica y tienen un número máximo prefijado de etiquetas lingüísticas. Sin embargo, el conocimiento evolucionado por Fuzzy-UCS es mucho más interpretable que el de UCS. Por ejemplo, en el problema *iris*, algunas de las reglas más generales y numerosas creadas por UCS son:

1. **SI** $x_1 \in$ [4,30, 7,76] y $x_2 \in$ [2,91, 4,40] y $x_3 \in$ [1,00, 3,77] y $x_4 \in$ [0,10, 2,19] **ENTONCES** Iris-setosa
2. **SI** $x_1 \in$ [4,30, 6,06] y $x_2 \in$ [2,00, 2,53] y $x_3 \in$ [4,30, 6,90] y $x_4 \in$ [0,10, 2,50] **ENTONCES** Iris-virginica
3. **SI** $x_1 \in$ [4,30, 7,71] y $x_2 \in$ [2,00, 3,90] y $x_3 \in$ [2,63, 6,90] y $x_4 \in$ [0,10, 1,39] **ENTONCES** Iris-versicolor

donde se define cada variable con un intervalo cuyos límites inferior y superior pueden tomar cualquier valor válido para el atributo. En cambio, considerando que las cinco etiquetas lingüísticas por variable son {XS, S, M, L, XL}, algunas de las reglas más numerosas y precisas evolucionadas por Fuzzy-UCS presentan la siguiente forma:

1. **SI** x_3 es {XS, S o L} y x_4 es {XS o S} **ENTONCES** Iris-setosa **CON** F =1
2. **SI** x_1 no es M y x_4 es {L o XL} **ENTONCES** Iris-virginica **CON** F =1
3. **SI** x_3 es M y x_4 no es L **ENTONCES** Iris-versicolor **CON** F =1

Cuadro 2: Tabla comparativa de la precisión en prueba de Fuzzy-UCS con SMO, UCS, C4.5, Fuzzy LogitBoost (Fuzzy LBoost) y Fuzzy GP. Por cada conjunto de entrenamiento, se marca en negrita el algoritmo con porcentaje de acierto de prueba más alto. La última fila recoge el *ranking* medio de cada algoritmo.

Problema	**SMO**	**UCS**	**C4.5**	**Fuzzy LBoost**	**Fuzzy GP**	**Fuzzy-UCS**
annealing	97,24 %	98,83 %	**98,90 %**	76,20 %	77,86 %	98,08 %
balance	86,89 %	78,11 %	77,42 %	80,93 %	69,73 %	**88,85 %**
bupa	58,28 %	**66,75 %**	62,31 %	65,97 %	56,62 %	61,99 %
glass	56,93 %	**69,34 %**	66,15 %	66,29 %	48,89 %	63,23 %
heart-c	**85,15 %**	81,46 %	78,45 %	57,19 %	73,98 %	83,48 %
heart-s	**84,44 %**	74,33 %	79,26 %	58,26 %	73,70 %	81,11 %
iris	**96,67 %**	95,13 %	94,00 %	96,00 %	94,47 %	95,40 %
wbcd	**96,85 %**	96,54 %	94,99 %	95,44 %	93,31 %	95,99 %
wine	**99,44 %**	94,89 %	93,89 %	81,59 %	82,91 %	94,89 %
zoo	96,50 %	**96,66 %**	92,81 %	41,89 %	71,18 %	96,27 %
Ranking	**2,3**	**2,45**	**3,8**	**4,3**	**5,4**	**2,75**

donde algunas de las variables han sido completamente generalizadas. En las reglas no se muestran los pesos por clase, pues son únicamente información interna que no es necesaria para inferir la clase. Véase que las reglas difusas generadas por Fuzzy-UCS pueden ser interpretadas más fácilmente que las de UCS, pues cada variable puede tomar un conjunto de cinco términos lingüísticos.

Finalmente, destacar que no todas las reglas evolucionadas se usan para la clasificación de los ejemplos de prueba. Actualmente se está trabajando en métodos de eliminado de reglas que no se consideran relevantes para la clasificación.

4. Conclusiones

Este artículo ha presentado Fuzzy-UCS, un sistema clasificador de la família de Michigan que evoluciona incrementalmente un conjunto de reglas difusas que cooperan para predecir la clase de ejemplos desconocidos. La experimentación realizada evidencia que Fuzzy-UCS es un sistema altamente competitivo con algunas técnicas bastamente usadas en la disciplina de aprendizaje supervisado, como son el C4.5, el SMO y el UCS. Además, el rendimiento supera ampliamente al presentado por otros métodos con representación difusa, como el Fuzzy GP y el Fuzzy LogitBoost.

El análisis experimental también ha servido para establecer las líneas de trabajo futuro. Se ha observado que no todas las reglas son necesarias para la inferencia de la clase de un ejemplo. Por lo tanto, en el trabajo futuro se desarrollarán técnicas para compactar la base de reglas con una disminución mínima, si no nula, de la precisión de prueba.

Agradecimientos

Los autores agradecen el soporte de *Enginyeria i Arquitectura La Salle*, Universidad Ramon Llull, y el soporte del *Ministerio de Ciencia y Tecnología* bajo los proyectos TIN2005-08386-C05-01 y TIN2005-08386-C05-04, y la *Generalitat de Catalunya* bajo las becas 2005FI-00252 y 2005SGR-00302.

Referencias

[1] J. Alcalá-Fdez, M.J. del Jesus, J.M. Garrell, F. Herrera, C. Herbás, and L. Sánchez. Proyecto KEEL: Desarrollo de una herramienta para el análisis e implementación de algoritmos de extracción de conocimiento evolutivos. In J.S. Aguilar

R. Giráldez, J.C. Riquelme, editor, *Tendencias de la Minería de Datos en España, Red Española de Minería de Datos y Aprendizage*, pages 413–424, 2004.

[2] E. Bernadó-Mansilla and J.M. Garrell. Accuracy-Based Learning Classifier Systems: Models, Analysis and Applications to Classification Tasks. *Evolutionary Computation*, 11(3):209–238, 2003.

[3] C.L Blake and C.J. Merz. *UCI Repository of ML Databases: http://www.ics.uc.edu/ mlearn/MLRepository.html.* Univ. of California, 1998.

[4] J. Casillas, B. Carse, and L. Bull. Fuzzy-XCS: an Accuracy-Based Fuzzy Classifier System. In *Proceedings of the XII Congreso Español sobre Tecnología y Lógica Fuzzy (ESTYLF 2004)*, pages 369–376, Jaen, Spain, 2004.

[5] O. Cordón, M.J. del Jesús, and F. Herrera. Analyzing the Reasoning Mechanisms in Fuzzy Rule Based Classification Systems. *Mathware & Soft Computing*, 5:321–332, 1998.

[6] D.E. Goldberg. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms.* Kluwer Academic Publishers, 1 edition, 2002.

[7] J.H. Holland. *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, 1975.

[8] J.H. Holland. Adaptation. In R. Rosen and F. Snell, editors, *Progress in Theoretical Biology*, volume 4, pages 263–293. New York: Academic Press, 1976.

[9] H. Ishibuchi and T. Yamamoto. Rule weight specification in fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems*, 13(4):428–435, 2005 .

[10] A. Orriols-Puig and E. Bernadó-Mansilla. *A Further Look at UCS Classifier System.* Springer, in press.

[11] J. Otero and L. Sánchez. Induction of descriptive fuzzy classifiers with the logitboost algorithm. *Soft Computing*, 10(9):825–835, 2006.

[12] J. Platt. Fast Training of Support Vector Machines using Sequential Minimal Opt. In *Adv. in Kernel Methods - Support Vector Lear.* MIT Press, 1998.

[13] J.R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, San Mateo, California, 1995.

[14] L. Sánchez and I. Couso. Learning with Imprecise Examples with GA-P Algorithms. *Soft Computing*, 5(1–4):305–319, 1998.

[15] T.G. Dietterich. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, 10(7):1895–1924, 1998.

[16] M. Valenzuela-Radón. The Fuzzy Classifier System: A Classifier System for Continuously Varying Variables. In *4th ICGA*, pages 346–353. Morgan Kaufmann, 1991.

[17] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer Verlag, New York, 1995.

[18] I.H Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, San Francisco, 2nd edition, 2005.

Actas de las I Jornadas sobre Algoritmos Evolutivos y Metaheurísticas

JAEM'07

Editadas por
Enrique Alba
Francisco Chicano
Francisco Herrera
Francisco Luna
Gabriel Luque
Antonio J. Nebro

Zaragoza, 12 y 13 de Septiembre de 2007