

# Construcción de árboles de decisión usando el Nivel de Impureza

José Ranilla Pastor    Marcos García Pellitero    Antonio Bahamonde  
Centro de Inteligencia Artificial. Universidad de Oviedo en Gijón  
Campus de Viesques. E-33271 Gijón, España  
e-mail: {ranilla, pelliter, antonio}@aic.uniovi.es  
<http://www.aic.uniovi.es>

**Palabras clave:** Aprendizaje a partir de Ejemplos, Árboles de Decisión, Nivel de Impureza.

**Keywords:** Machine Learning from Examples, Decision Trees, Impurity Level.

**RESUMEN.** *El nivel de impureza es una medida de la calidad de la clasificación de reglas aprendidas. En este artículo exploramos sus usos como un heurístico para encontrar tests razonables con los que construir árboles de decisión a partir de un conjunto de ejemplos de entrenamiento. Para hacer esto, implementamos un algoritmo que sigue los pasos del C4.5 de Quinlan y que toma prestado de un sistema anterior nuestro, ABANICO, los métodos de poda para obtener reglas generales y precisas a partir de los árboles así obtenidos. El resultado es un nuevo algoritmo de aprendizaje con una precisión ligeramente superior a la del C4.5 pero que produce un número considerablemente menor de reglas.*

**ABSTRACT.** *The impurity level is a measure of the classification quality of learned rules. In this paper we explore their uses as a heuristic to find reasonable tests to build decision trees from a set of training examples. To do this, we implemented an algorithm following the steps of Quinlan's C4.5, and we borrowed from our previous learning system ABANICO the pruning methods to obtain general and accurate rules from trees so obtained. The result is a new learning algorithm with an accuracy level slightly better than C4.5's, but producing a considerably fewer number of rules.*

## 1.- INTRODUCCIÓN.

En el proceso de construcción de las estructuras de clasificación que se están aprendiendo a partir de ejemplos, es conveniente disponer de una medida que permita establecer una estimación de lo ajustadas que serán las predicciones sobre los casos no vistos. Una de estas medidas es, obviamente, la usada por Quinlan en (Quinlan 93) para su sistema C4.5 que se muestra especialmente eficaz cuando el resultado es un árbol de decisión. Ahora bien, si se desean sintetizar reglas de clasificación surgen diversos problemas que obligan a utilizar estrategias o métricas distintas a las planteadas para la construcción del árbol, lo que redundará en la pérdida de la claridad conceptual y, probablemente, en la eficacia del conjunto de clasificadores sintetizado.

La alternativa que aquí proponemos está basada en una métrica, denominada *Nivel de Impureza* (Ranilla, Mones, Bahamonde 97), que es una medida heurística sobre la calidad de las reglas aprendidas a partir de un conjunto de ejemplos y que pretendemos extender su uso al ámbito de la construcción de los árboles de decisión.

Para comprobar la eficacia de esta propuesta hemos implementado un algoritmo de aprendizaje a partir de ejemplos que sigue unos pasos paralelos a los del C4.5 de Quinlan en lo que se refiere a la generación del árbol, pero donde hemos sustituido las fórmulas de la incertidumbre que allí se usan, por generalizaciones evidentes donde ahora es el nivel de impureza la magnitud protagonista. Los árboles que se obtienen de cualquiera de estas maneras, se suelen someter a un

proceso de poda en el que se trata de evitar los problemas del sobreajuste a los conjuntos de entrenamiento. Pues bien, antes de realizar estas podas, los árboles generados con la ayuda del nivel de impureza dan unos resultados de precisión en la clasificación que son ligeramente mejores que los obtenidos por Quinlan. Sin embargo, Quinlan consigue que estos árboles sean más pequeños que los nuestros.

De todas formas, la fuerza de todos estos sistemas debe buscarse en las versiones donde se poda el resultado de la primera inducción. Aquí los caminos seguidos por el C4.5 y nuestro sistema divergen radicalmente. Nosotros tomamos prestado de nuestro anterior sistema ABANICO (Ranilla 98) los mecanismos de poda que ahora aplicamos a las ramas de los árboles previos leídas como reglas que concluyen las clases que etiquetan sus hojas. De esta forma conseguimos reducir y generalizar el conocimiento sintetizado en la anterior fase de inducción y tenemos un reducidísimo conjunto de reglas que se aplican por mínima distancia, lo que les da una gran flexibilidad y facilita el uso sobre casos no vistos de una manera natural.

En el artículo, tras presentar los algoritmos de construcción de nuestros árboles y los mecanismos de poda, se termina con una sección en la que mostramos los resultados de las pruebas experimentales realizadas para justificar, de esta manera, las ventajas del algoritmo propuesto.

## 2.- EL NIVEL DE IMPUREZA DE UNA REGLA DE CLASIFICACIÓN.

Cuando se ha de elegir entre distintas reglas es deseable utilizar un criterio de evaluación que nos lleve a seleccionar aquellas reglas que, previsiblemente, aporten un mayor número de aciertos en la clasificación de los ejemplos no vistos durante el periodo de aprendizaje.

El criterio más elemental es considerar el número de aciertos y fallos de la regla sobre los propios ejemplos de entrenamiento. En este sentido una regla sería tanto mejor cuantos más aciertos tenga, pero para poder comparar la calidad de dos reglas sería más justo considerar, para cada regla, la proporción de aciertos sobre el conjunto de ejemplos a los que se puede aplicar.

Sin embargo, en (Ranilla 98) se demuestra que este criterio no es suficientemente justo ya que no contabiliza el número de veces que la regla ha sido usada; dos reglas que siempre aciertan y que una se aplica en 2 ocasiones y la otra en 29 veces no se debería considerar que tienen igual calidad. Además, cuando en el conjunto de entrenamiento hay *ruido* la situación se complica. Una regla que tenga algún fallo en este conjunto de ejemplos no necesariamente tiene que ser peor que otra que no los tenga.

Teniendo en cuenta las consideraciones anteriores, la fórmula que finalmente proponemos para el cálculo del nivel de impureza está inspirada en el criterio que Aha (Aha 90) usa en el algoritmo IB3 para seleccionar los ejemplares más adecuados. En primer lugar, dada una regla que es aplicable  $n$  veces de las cuales tiene  $\#$ aciertos éxitos de clasificación, se determina el intervalo de confianza de su probabilidad de acierto. Para ello se usa la siguiente fórmula tomada de (Spiegel 70; p.162):

$$\frac{p + \frac{z^2}{2n} \pm z \sqrt{\frac{p(1-p)}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}}$$

$n$ : número de veces que se usó la regla,

$p$ : proporción de aciertos,

$z$ : un valor tomado de la tabla de la distribución normal según el nivel de confianza *alfa* usado para establecer el intervalo.

Una vez que se han calculado estos intervalos de confianza para las probabilidades de acierto de cada regla, se tendrán en cuenta las *peculiaridades de las clases* que concluyen cada una de las reglas. De esta forma podremos establecer preferencias entre reglas de distintas clases. En este caso, para mantener la justicia en la evaluación de los distintos resultados, hay que considerar la dificultad de acertar en la clasificación en cada clase.

DEFINICIÓN.- Si  $C$  es una clase, llamaremos *regla canónica del azar de esa clase  $C$*  a la que la tiene por conclusión sin ningún antecedente o condición. En símbolos:

$C \leftarrow$

DEFINICIÓN.- Sea  $R$  una regla de la clase  $C$  que tiene un intervalo de confianza de la probabilidad de acierto [**Extremo\_izquierdo, Extremo\_derecho**]. Sea  $C \leftarrow$  la regla canónica del azar de esa clase. Sea [**Azar\_Extremo\_izquierdo, Azar\_Extremo\_derecho**] el intervalo de confianza de acierto de esta regla canónica. Entonces, se define el *nivel de impureza de  $R$*  como el nivel de solapamiento de estos dos intervalos. En fórmula,

$$\text{nivel - impureza}(R) = 100 * \frac{\text{Azar\_extremo\_derecho} - \text{Extremo\_izquierdo}}{\text{Extremo\_derecho} - \text{Extremo\_izquierdo}}$$

Intuitivamente este nivel de solapamiento mide el porcentaje del intervalo de confianza de  $R$  que está solapado por el de la regla canónica o la regla de la clasificación al azar. Pero no siempre será cero el umbral de impureza más adecuado para seleccionar un conjunto de clasificadores. Si este nivel es negativo tendríamos reglas con intervalo de confianza por encima del azar. La relación de preferencia entre reglas ahora es clara:

***Una regla se preferirá a otra si y solo si su nivel de impureza es menor***

En las secciones siguientes se verá cómo usar este criterio de preferencia para elegir el corte más adecuado en el cálculo de los nodos del árbol de decisión, así como para obtener el conjunto final de reglas de clasificación más prometedor.

### 3.- EL NIVEL DE IMPUREZA EN LA CONSTRUCCIÓN DE ÁRBOLES.

Seguramente el algoritmo de aprendizaje automático a partir de ejemplos más popular es el ID3 de Quinlan. En (Quinlan 93; cap. 2) se explica que la idea central del algoritmo ID3 y de otros como CART, PLS1, ASSISTANT 86 o el OC1, está en los trabajos de Hunt que dieron lugar al CLS (*Concept Learning System*) en los años 60.

Para describir este esqueleto de algoritmo de Hunt hay que presentar primero los árboles de decisión. Se trata de árboles donde los nodos están etiquetados por un test relativo a los valores de los ejemplos en un atributo (o varios incluso) y las hojas por una de las clases. Para precisar el método de construcción consideremos un conjunto de ejemplos de entrenamiento  $E$  que puede pertenecer a alguna de las clases del conjunto  $\{C_1, C_2, \dots, C_k\}$ . Entonces caben tres posibilidades:

- $E$  está formado por ejemplos que pertenecen todos a una misma clase  $C_i$ . El árbol es entonces una hoja etiquetada por  $C_i$ .
- $E$  es vacío en cuyo caso el árbol será una hoja etiquetada por alguna clase elegida por algún criterio heurístico. En C4.5 se elige la clase más abundante en los ejemplos del nodo padre.
- $E$  contiene ejemplos de varias clases. En este caso se busca un test que divida el conjunto  $E$  en subconjuntos donde los ejemplos o bien pertenezcan todos a una sola clase o estén próximos a ello. En la elección de este test es donde se diferencian los algoritmos de construcción de árboles de decisión concretos. En general si el test aplicado a cada uno de los ejemplos tiene como respuestas posibles el conjunto  $\{Res_1, \dots, Res_m\}$ , se etiqueta el nodo con el test y se construyen  $m$  ramas cada una de ellas terminando en nodos con los subconjuntos de entrenamiento que tienen cada una de las respuestas posibles:  $E_1, \dots, E_m$ , como se muestra en la Figura 1. El árbol de decisión para  $E$  se completa aplicando recursivamente este algoritmo a cada uno de los conjuntos  $E_i$  anteriores.

En el caso concreto de ID3 la medida usada para la elección del test más prometedor es la *incertidumbre*, que se define como

$$H(p_1, p_2, \dots, p_m) = - \sum_{i=1}^m p_i * \log_2(p_i),$$

donde  $(p_1, p_2, \dots, p_m)$  son una distribución de probabilidad discreta.

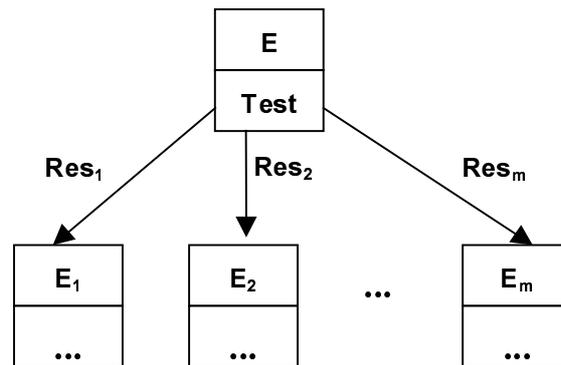


Figura 1. Árbol de decisión para un test sobre un conjunto de ejemplos con varias clases.

Una vez que se ha presentado el nivel de impureza de una regla de clasificación y los principios básicos de Hunt para la construcción de árboles de decisión, en el resto de esta sección se verá como utilizar dicha medida para obtener árboles de decisión. El proceso de construcción es exactamente igual al utilizado por Quinlan para sus sistemas ID3 y C4.5 sustituyendo, obviamente, la incertidumbre por el nivel de impureza. Al sistema resultante lo hemos bautizado con el nombre de ARNI: *Árboles y Reglas basados en el Nivel de Impureza*.

El primer paso cuando se está analizando el test más adecuado para un nuevo nodo del árbol es determinar cual es el atributo que proporciona una mejor valoración. En ARNI, al igual que en la mayoría de los sistemas de este tipo, para cada atributo candidato se proponen tantos nodos como tests sean posibles según su tipo, y se distribuyen los ejemplos. Evidentemente, en cada rama caerán los ejemplos que satisfacen todos los tests generados hasta el momento. El cálculo del Nivel de Impureza para cada rama se realiza suponiendo que de ella cuelga una hoja con la clase más abundante de los ejemplos asociados, lo que posibilita disponer de la información necesaria para determinar el intervalo de confianza de las reglas canónicas del azar. A continuación, se hace la media ponderada de cada rama obteniendo el nivel de impureza medio asociado al atributo. Finalmente, el atributo seleccionado será el de menor valor para su nivel de impureza medio.

Las expresiones utilizadas para el cálculo de los valores medios según el tipo del atributo son:

- **Simbólicos.** Sea el atributo  $X$  el que está siendo evaluado, y  $E$  el conjunto de ejemplos. Por ser  $X$  de tipo simbólico, se generan tantas ramas como valores pueda tomar el atributo, dividiéndose  $E$  en  $E_1, \dots, E_m$ . El nivel de impureza de  $X$  se calcularía con la expresión:

$$NI(E, X) = \sum_{i=1}^m \frac{|E_i|}{|E|} * Nivel\_Impureza(E_i)$$

- **Numéricos.** En este caso, el test se va a caracterizar por un valor de corte que generará dos ramas. La primera se corresponderá con los valores menores o iguales al corte, y la segunda con los valores superiores. El problema está entonces en determinar el corte que produzca un nivel de impureza menor. Para su cálculo se ordenan, de menor a mayor, los valores que toman los ejemplos en el atributo y se toma como valor inicial una cantidad inferior al menor valor real. A continuación, se inicia un recorrido y para cada par consecutivo de ejemplos se considera como valor del corte al punto intermedio de los valores de los ejemplos en el atributo. Cuando los valores de dos ejemplos consecutivos en el atributo son iguales se sigue avanzando hasta localizar un par de valores distintos. Así, cada posible corte tendrá un nivel de impureza dado por

$$NI(E, X) = \text{minimo}\{NI\_Corte(E, X, \alpha) : \alpha \text{ cortes de los valores de } X\},$$

siendo

$$NI\_Corte(E, X, \alpha) = \sum_{i=1}^2 \frac{|E_i|}{|E|} * Nivel\_Impureza(E_i)$$

y donde  $E_1$  será el subconjunto de ejemplos con valores de  $X$  menores o iguales que  $\alpha$  y  $E_2$  los de valores superiores. El corte que obtenga un menor valor en la expresión anterior será el elegido, y dicho valor del nivel de impureza será el representante del atributo para competir con el resto de atributos.

El último aspecto que falta por precisar es el tratamiento de los *valores desconocidos* (missing values). Como en C4.5, al plantear un test sobre un atributo con valores desconocidos solamente se asignan a las posibles ramas los ejemplos que no tienen valores desconocidos y, además, se añade una rama *ficticia* con los ejemplos restantes (los que presentan missing values), que será tratada igual que el resto para el cálculo del nivel de impureza.

Si finalmente el test elegido para el nodo tiene valores desconocidos en algún ejemplo, se utiliza un esquema de probabilidades o pesos. Si el ejemplo no tiene valores desconocidos el peso será 1 y, en caso contrario, será la probabilidad de la rama. La fórmula utilizada para la propagación de estos pesos es

$$W * \frac{|E_i|}{|E|},$$

donde  $E$  es el conjunto de ejemplos,  $W$  el peso asociado a  $E$  y  $E_i$  cada una de las particiones generadas por el test.

#### 4.- OBTENCIÓN DE UN CONJUNTO DE REGLAS A PARTIR DEL ÁRBOL.

Una vez calculado el árbol de decisión, para obtener el conjunto final de reglas que sintetiza ARNI se lleva a cabo el siguiente proceso. Partiendo del árbol sin podar, se realiza un recorrido en profundidad de la raíz a las hojas. Cada rama así calculada constituye una regla candidata, donde

- la etiqueta de la hoja será la clase y
- las etiquetas o *test* de los nodos intermedios, incluida la de la raíz del árbol, serán los antecedentes de la regla.

Debido a que los árboles de decisión se construyen con el ánimo de hacer un cubrimiento total del espacio, puede ocurrir, sobre todo en problemas de tipo simbólico, que el conjunto de ejemplos a los cuales es aplicable una rama sea vacío. Cuando se detecta esta situación no se generará la regla asociada. Por otra parte, cuando en una rama hay varios tests sobre el mismo atributo y este es de tipo numérico, únicamente se genera un antecedente para la regla, cuyo valor será el intervalo acotado por el mínimo y máximo de los tests asociados.

Este conjunto inicial, que es equivalente al árbol dado que clasifica los mismos ejemplos y de igual forma, se somete a un proceso de mejora, tanto en calidad como en cantidad, similar al utilizado por el sistema ABANICO (Ranilla 98) y que hace un uso exhaustivo del nivel de impureza. Para ilustrar la explicación vamos a considerar que las reglas de este conjunto inicial están descritas por una clase y una lista de pares atributo valor como se muestra en el siguiente ejemplo

**[Regla]      [clase; (atributo-1, valor-1), . . . , (atributo-n, valor-n)],**

donde los *valor-i* no tienen porque ser los valores originales y, en el caso numérico, serán intervalos; es decir, en general se trata de conjuntos de valores.

Pues bien, para cada regla de este conjunto inicial se aplica un mecanismo, denominado en ABANICO *calificación*, cuyos pasos se enumeran a continuación.

1. Asociar a cada par (**atributo-i, valor-i**) el nivel de impureza de la regla

**clase ← ((atributo-i, valor-i))**

2. A continuación se ordenan la lista de elementos de la regla por el menor nivel de impureza. Para no complicar la notación supondremos que **[Regla]** ya tiene ordenados sus descriptores. Esto permite plantear la posibilidad de considerar las reglas

**clase** ← ((atributo-1, valor-1))

**clase** ← ((atributo-1, valor-1) (atributo-2, valor-2))

**clase** ← ((atributo-1, valor-1) (atributo-2, valor-2) (atributo-3, valor-3))

.....

Esta lista se terminará cuando o bien se encuentre una descripción que no tenga fallos o bien que se haya agotado la lista de elementos posibles del ejemplo de partida. Cada una de estas posibles reglas es filtrada rechazándose aquellas que no tengan una calidad suficiente. Las reglas que superen el filtro se añaden a una lista de reglas provisionales que es ordenada por los menores niveles de impureza.

3. Sobre la mejor regla, la de menor nivel de impureza, obtenida en el paso anterior para el prototipo inicial, se tratará de borrar algún elemento redundante. En el paso anterior, el último antecedente considerado fue introducido para tratar de mejorar el rendimiento de la regla sin él, pero una vez que nos vemos obligados a admitir su presencia en la regla, puede que alguno de los elementos anteriores ya no sea necesario. Así, se revisan las reglas que resulten de prescindir, de la regla de referencia, de cada uno de los elementos anteriores de la regla (del penúltimo al primero). Cuando alguno de ellos resulte prescindible continuaremos el proceso sin él. El resultado de este proceso se añade también a la lista de reglas provisionales.
4. Por último, la lista de reglas provisionales es sometida, a su vez, a un último filtro. Sólo se aceptarán las reglas cuyo nivel de impureza sea, a lo sumo, 10 puntos peor que el de la mejor regla de la lista.

Concluida la fase de cualificación, el conjunto obtenido es sometido a un proceso de *selección* que de forma sintetizada consta de las etapas siguientes.

1. El conjunto de reglas es filtrado de tal forma que, en caso de problemas con *ruido*, se eliminen las reglas que clasifican a muy pocos ejemplos. Obviamente, el filtrado únicamente tendrá efecto en aquellos problemas en los que no es posible encontrar, en todos los casos, reglas buenas sin errores internos.
2. Seguidamente las reglas vuelven a ser ordenadas por su nivel de impureza y, el subconjunto de reglas elegido, resultará de fijar un nivel de impureza y quedarse con las reglas que tengan un nivel mejor (más bajo). Para fijar este corte se recorre la lista de los niveles de impureza disponibles y, con cada opción, determinamos un conjunto de reglas que tendrá una valoración. Obviamente, el conjunto seleccionado será aquel que tenga la mejor valoración.
3. En nuevo conjunto de reglas sufren un proceso de eliminación de redundancias simétrico al descrito en la cualificación. Puede haber reglas seleccionadas que no representen ninguna mejora en el número de aciertos sobre el conjunto de ejemplos de entrenamiento, incluso que empeoren este resultado. Por eso se explora la posibilidad de prescindir de cada regla empezando por la última (por nivel de impureza) hasta la primera. Si alguna de ellas es prescindible se continúa el proceso sin ella.
4. Finalmente, la selección implementa un proceso para decidir sobre la inclusión o no de una *regla por defecto* cuando estamos frente a problemas en los cuales todos los atributos son simbólicos.

## 5.- PRUEBAS EXPERIMENTALES.

En esta sección se recogen los resultados experimentales obtenidos por los sistemas ARNI y C4.5 (Quinlan 93), tanto para las versiones basadas en árboles de decisión como para las variantes que producen reglas de clasificación.

El grupo de problemas que se ha seleccionado está compuesto por 16 conjuntos de ejemplos, conocidos como *los problemas de Holte*, usados por Holte en (Holte 93) para comparar su sistema 1R con el C4.5 y que, en la actualidad, se está convirtiendo en un estándar de referente mundial. Todos los conjuntos se han utilizado en su versión original y se han descargado del almacén de la Universidad de California en Irvine (Blake *et al.* 98).

Por otra parte, el tipo de experimento realizado ha sido *validación cruzada estratificada*, con 10 particiones y repitiendo cada experimento 5 veces. Esto significa que se ha dividido cada conjunto de forma aleatoria en 10 partes aproximadamente iguales, manteniendo la proporción de cada clase en el conjunto inicial. A continuación y, de forma iterativa, se utiliza cada partición como conjunto de prueba sobre las estructuras de clasificación aprendidas con las particiones restantes. Este mecanismo se repite cinco veces, dando lugar a un total de 50 experimentos para cada conjunto.

Todo el proceso experimental se ha efectuado con las herramientas que proporciona la librería MLC++ (Kohavi 94), en el mismo entorno informático y con la misma semilla para el generador de números aleatorios de la MLC++, lo que garantiza que las particiones obtenidas para cada problema son las mismas para todos los sistemas.

En la siguiente tabla se recogen los resultados obtenidos con C4.5 y ARNI por las variantes que concluyen árboles de decisión sin realizar el proceso de *poda*. Como se puede observar, la precisión de ARNI es, en media, ligeramente mejor; aunque si se realiza un *contraste t con dos colas* considerando como hipótesis nula que los dos sistemas tienen medias iguales, se obtiene un nivel de significación (probabilidad de error de rechazo de la hipótesis nula) del 0.766.

	C4.5	ARNI	C4.5	ARNI
	Error (%)	Error (%)	Nodos	Nodos
<b>BC</b>	33.17 ± 1.08	33.97 ± 1.11	171.64	224.98
<b>CH</b>	1.06 ± 0.08	1.38 ± 0.08	83.80	98.52
<b>G2</b>	23.90 ± 1.34	18.68 ± 1.31	24.40	30.32
<b>GL</b>	33.79 ± 1.36	30.52 ± 1.51	50.32	51.76
<b>HD</b>	23.20 ± 1.02	26.13 ± 1.16	74.16	64.76
<b>HE</b>	21.18 ± 1.66	18.04 ± 1.62	30.04	49.80
<b>HO</b>	17.95 ± 0.83	16.31 ± 0.78	101.82	148.00
<b>HY</b>	0.78 ± 0.07	1.02 ± 0.06	32.60	39.76
<b>IR</b>	5.47 ± 0.85	5.20 ± 0.79	8.60	8.84
<b>LA</b>	21.73 ± 1.86	21.67 ± 2.27	15.32	21.06
<b>LY</b>	23.78 ± 1.80	29.21 ± 1.92	39.72	45.10
<b>MU</b>	0.00 ± 0.00	0.00 ± 0.00	30.14	30.74
<b>SE</b>	2.47 ± 0.14	2.57 ± 0.13	92.76	149.00
<b>SO</b>	2.60 ± 1.01	0.80 ± 0.80	8.40	8.00
<b>VO</b>	4.65 ± 0.44	5.48 ± 0.48	26.16	59.72
<b>V1</b>	10.36 ± 0.56	12.05 ± 0.59	48.80	90.68
Media	14.13	13.94	52.42	70.07

**Tabla 1.** Error medio ± error estándar alcanzado en los experimentos, así como el número medio de nodos de los árboles.

Respecto al número medio de nodos y, utilizando el mismo tipo de contraste pero ahora con *una cola*, el sistema C4.5 sintetiza árboles de clasificación significativamente más compactos (con un nivel de significación próximo al 0,0025).

Para las variantes que producen reglas de clasificación, en el caso del C4.5 vía los procesos de *poda* y conversión descritos en (Quinlan 87) y (Quinlan 93) y para ARNI mediante el mecanismo descrito en la sección 4, los resultados se muestran en la Tabla 2. Desde el punto de vista del error medio las diferencias siguen sin ser significativas, aunque la diferencia entre los valores medios del error es algo más favorable para ARNI que la de los árboles de decisión.

Ahora bien, si nos fijamos en el número medio de reglas/antecedentes se observa que ahora el conjunto de reglas obtenido por ARNI es significativamente mejor (en este caso el nivel de significación es 0,005), más compacto.

	C4.5	ARNI	C4.5		ARNI	
	Error (%)	Error (%)	Reglas	Antecedentes	Reglas	Antecedentes
<b>BC</b>	29.52 ± 1.26	32.58 ± 1.18	8.66	19.16	12.96	30.18
<b>CH</b>	1.02 ± 0.08	2.10 ± 0.10	26.70	99.40	8.92	36.36
<b>G2</b>	22.44 ± 1.39	18.45 ± 1.43	8.32	21.62	6.18	11.18
<b>GL</b>	33.97 ± 1.26	32.72 ± 1.57	13.72	48.92	13.84	49.78
<b>HD</b>	20.37 ± 1.15	20.62 ± 1.22	13.46	37.08	7.36	17.14
<b>HE</b>	21.43 ± 1.92	18.58 ± 1.46	8.36	21.94	5.56	15.96
<b>HO</b>	16.96 ± 0.76	15.92 ± 0.82	5.74	10.74	3.58	9.02
<b>HY</b>	0.78 ± 0.06	0.87 ± 0.08	6.28	12.72	3.60	9.88
<b>IR</b>	4.80 ± 0.76	5.47 ± 0.85	4.02	6.04	3.84	5.96
<b>LA</b>	18.00 ± 1.87	17.67 ± 1.93	4.08	6.08	2.86	5.64
<b>LY</b>	23.79 ± 1.74	23.26 ± 1.61	10.16	23.12	7.66	14.28
<b>MU</b>	0.01 ± 0.00	1.59 ± 0.06	16.84	25.54	4.34	3.68
<b>SE</b>	2.30 ± 0.12	2.14 ± 0.13	12.54	40.14	2.42	7.48
<b>SO</b>	2.60 ± 1.01	0.80 ± 0.80	4.00	5.20	5.00	7.00
<b>VO</b>	4.83 ± 0.45	5.01 ± 0.47	6.20	13.96	2.38	4.86
<b>VI</b>	9.94 ± 0.65	10.90 ± 0.58	11.16	28.8	5.06	10.34
<b>Media</b>	13.30	13.04	10.01	26.28	5.97	14.92

**Tabla 2.** El error medio ± error estándar alcanzado en los experimentos, así como el número medio de reglas y número medio de antecedentes.

## BIBLIOGRAFÍA.

- Aha, D.W. (1990). *A Study of Instance-based Algorithms for Supervised Learning Tasks: Mathematical, Empirical and Psychological Evaluations*. Ph. D. Dissertation. Irvine, University of California.
- Blake, C., Keogh, E. & Merz, C.J. (1998). UCI repository of machine learning databases [<http://www.ics.uci.edu/~mllearn/MLRepository.html>]. Irvine, CA: University of California, Department of Information and Computer Science.
- Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, C. J. (1984). *Classification and Regression Trees*. Belmont, Wadsworth International Group.
- Brodley, C.E. & Utgoff, P.E. (1995). Multivariate decision trees. *Machine Learning*, 19:45-77.
- Holte, R.C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63-91.
- Kohavi, R., John, G., Long, R. & Manley, D. (1994). MLC++: A machine learning library in C++. In *Proceeding of Sixth International Conference on Tools with Artificial Intelligence*, IEEE Computer Society Press, 740-743.
- Quinlan, J.R. (1979). Discovering rules from large collections of examples: a case study. In Michie, D. (ed.): *Expert Systems in the Micro-Electronic Age*, Edinburgh University Press, 168-201.
- Quinlan, J.R. (1987). Simplifying Decision Trees. *International Journal of Man-Machine Studies*, 27:221-234.
- Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, California.
- Ranilla, J. (1998). *ABANICO: Aprendizaje Basado en Agrupación Numérica en Intervalos COntinuos*. Tesis doctoral dirigida por A. Bahamonde. Departamento de Informática, Universidad de Oviedo.
- Ranilla, J., Mones, R. & Bahamonde, A. (1997). El Nivel de Impureza de una Regla de Clasificación Aprendida a partir de Ejemplos. *Actas de la VII Conferencia de la Asociación Española para la Inteligencia Artificial*, Torremolinos, Málaga, 479-488. También en *Revista Iberoamericana de la Inteligencia Artificial* (4:4-11, 1998) y en *Novatica* (131:37-43, 1998).
- Spiegel, M.R. (1970). *Estadística*. McGraw-Hill, Atlacomulco, México.