INGENIERÍA EN SISTEMAS COMPUTACIONALES

INTELIGENCIA ARTIFICIAL

~ ALGORITMO C4.5 ~

ING. BRUNO LÓPEZ TAKEYAS

ALUMNOS:

José Antonio Espino López
Javier Eduardo Tijerina Flores
Manuel Cedano Mendoza
Eleazar de la Fuente Amaya
Juan José Pérez González
Aníbal Chiñas Carballo

Nuevo Laredo, Tamaulipas, Noviembre del 2005

ÍNDICE

INTRODUCCIÓN

LA FAMILIA TDIDT

La familia de los Top Down Induction Trees (TDIDT) pertenece a los métodos inductivos del Aprendizaje Automático que aprenden a partir de ejemplos preclasificados. En Minería de Datos, se utiliza para modelar las clasificaciones en los datos mediante árboles de decisión.

CONSTRUCCIÓN DE LOS ÁRBOLES DE DECISIÓN

Los árboles TDIDT, a los cuales pertenecen los generados por el ID3 y pos el C4.5, se construyen a partir del método de Hunt. El esqueleto de este método para construir un árbol de decisión a partir de un conjunto T de datos de entrenamiento es muy simple. Sean las clases {C1, C2,. . ., Ck}. Existen tres posibilidades:

1. T contiene uno o más casos, todos pertenecientes a una única clase Cj:
   El árbol de decisión para T es una hoja identificando la clase Cj.
2. T no contiene ningún caso:
   El árbol de decisión es una hoja, pero la clase asociada debe ser determinada por información que no pertenece a T. Por ejemplo, una hoja puede escogerse de acuerdo a conocimientos de base del dominio, como ser la clase mayoritaria.
3. T contiene casos pertenecientes a varias clases:
   En este caso, la idea es refinar T en subconjuntos de casos que tiendan, o parezcan tender hacia una colección de casos pertenecientes a una única clase. Se elige una prueba basada en un único

ALGORITMO C4.5 ORIGEN

El algoritmo c4.5 fue desarrollado por JR Quinlan en 1993, como una extensión (mejora) del algoritmo ID3 que desarrollo en 1986.

El algoritmo C4.5 genera un árbol de decisión a partir de los datos mediante particiones realizadas recursivamente. El árbol se construye mediante la estrategia de profundidad-primero *(depth-first).*

El algoritmo considera todas las pruebas posibles que pueden dividir el conjunto de datos y selecciona la prueba que resulta en la mayor ganancia de información. Para cada atributo discreto, se considera una prueba con *n* resultados, siendo *n* el número de valores posibles que puede tomar el atributo. Para cada atributo continuo, se realiza una *prueba binaria* sobre cada uno de los valores que toma el atributo en los datos. En cada nodo, el sistema debe decidir cuál prueba escoge para dividir los datos.

Los tres tipos de pruebas posibles propuestas por el C4.5 son:

La prueba "estándar" para las variables discretas, con un resultado y una rama para cada valor posible de la variable.
Una prueba más compleja, basada en una variable discreta, en donde los valores posibles son asignados a un número variable de grupos con un resultado posible para cada grupo, en lugar de para cada valor.
Si una variable A tiene valores numéricos continuos, se realiza una prueba binaria con resultados A <= Z y A > Z, para lo cual debe determinarse el valor límite *Z.*

Todas estas pruebas se evalúan de la misma manera, mirando el resultado de la proporción de ganancia, o alternativamente, el de la ganancia resultante de la división que producen. Ha sido útil agregar una restricción adicional: para cualquier división, al menos dos de los subconjuntos Ci deben contener un número razonable de casos. Esta restricción, que evita las subdivisiones casi triviales, es tenida en cuenta solamente cuando el conjunto C es pequeño.

CARACTERÍSTICAS DEL ALGORITMO C4.5

- Permite trabajar con valores continuos para los atributos, separando los posibles resultados en 2 ramas Ai<=N y Ai>N.
- Los árboles son menos frondosos, ya que cada hoja cubre una distribución de clases no una clase en particular.
- Utiliza el método "divide y vencerás" para generar el árbol de decisión inicial a partir de un conjunto de datos de entrenamiento.
- Se basa en la utilización del criterio de proporción de ganancia (gain ratio), definido como I(Xi,C)/H(Xi). De esta manera se consigue evitar que las variables con mayor número de posibles valores salgan beneficiadas en la selección.
- Es Recursivo.

HEURÍSTICA

Utiliza una técnica conocida como Gain Ratio (proporción de ganancia). Es una medida basada en información que considera diferentes números (y diferentes probabilidades) de los resultados de las pruebas.

ATRIBUTOS

Atributos de valores continuos: Inicialmente el algoritmo ID3 se planteó para atributos que presentaban un número discreto de valores. Podemos fácilmente incorporar atributos con valores continuos, simplemente dividiendo estos valores en intervalos discretos, de forma que el atributo tendrá siempre valores comprendidos en uno de estos intervalos.

Medidas alternativas en la selección de atributos: Al utilizar la ganancia de información estamos introduciendo involuntariamente un sesgo que favorece a los atributos con muchos valores distintos. Debido a que dividen el conjunto de ejemplos en muchos subconjuntos, la ganancia de información es forzosamente alta. Sin embargo, estos atributos no son buenos predictores de la función objetivo para nuevos ejemplos. Una medida alternativa que se ha usado con éxito es la "gain ratio".

Atributos con valores perdidos: En ciertos casos existen atributos de los cuales conocemos su valor para algunos ejemplos, y para otros no. Por ejemplo una base de datos médica en la que no a todos los pacientes se les ha practicado un análisis de sangre. En estos casos lo más común es estimar el valor basándose en otros ejemplos de los que sí conocemos el valor. Normalmente se fija la atención en los demás ejemplos de ese mismo nodo. Así, al ejemplo de valor desconocido se le da el valor que más aparezca en los demás ejemplos.

Atributos con pesos diferentes: En algunas tareas de aprendizaje los atributos pueden tener costes asociados. Por ejemplo, en una aplicación médica para diagnosticar enfermedades podemos tener atributos como temperatura, resultado de la biopsia, pulso, análisis de sangre, etc., que varían significativamente en su coste, monetario y relativo a molestias para el paciente.
Ventajas respecto al algoritmo ID3

MEJORAS DEL ALGORITMO C4.5

- Evitar Sobreajuste de los datos.
- Determinar que tan profundo debe crecer el árbol de decisión.
- Reducir errores en la poda.
- Condicionar la Post-Poda.
- Manejar atributos continuos.
- Escoger un rango de medida apropiado.
- Manejo de datos de entrenamiento con valores faltantes.
- Manejar atributos con diferentes valores.
- Mejorar la eficiencia computacional.

SOBREAJUSTE (OVERFITTING)

A medida que se añaden niveles AD, las hipótesis se refinan tanto que describen muy bien los ejemplos utilizados en el aprendizaje, pero el error de clasificación puede aumentar al evaluar los ejemplos. Es decir, clasifica muy bien los datos de entrenamiento pero luego no sabe generalizar al conjunto de prueba. Es debido a que aprende hasta el ruido del conjunto de entrenamiento, adaptándose a las regularidades del conjunto de entrenamiento.

Este efecto es, por supuesto, indeseado. Hay varias causas posibles para que esto ocurra. Las principales son:

- Exceso de ruido (lo que se traduce en nodos adicionales)
- Un conjunto de entrenamiento demasiado pequeño como para ser una muestra representativa de la verdadera función objetivo.

Hay varias estrategias para evitar el sobreajuste en los datos. Pueden ser agrupadas en dos clases:

- Estrategias que frenan el crecimiento del árbol antes de que llegue a clasificar perfectamente los ejemplos del conjunto de entrenamiento.
- Estrategias que permiten que el árbol crezca completamente, y después realizan una poda.

POST PRUNNING (POST PODA)

Es una variante de la poda y es usada por el C4.5. Consiste en una vez generado el árbol completo, plantearse qué es lo que se debe "podar" para mejorar el rendimiento y de paso obtener un árbol más corto.
Pero además el C4.5 convierte el árbol a un conjunto de reglas antes de podarlo. Hay tres razones principales para hacer esto:

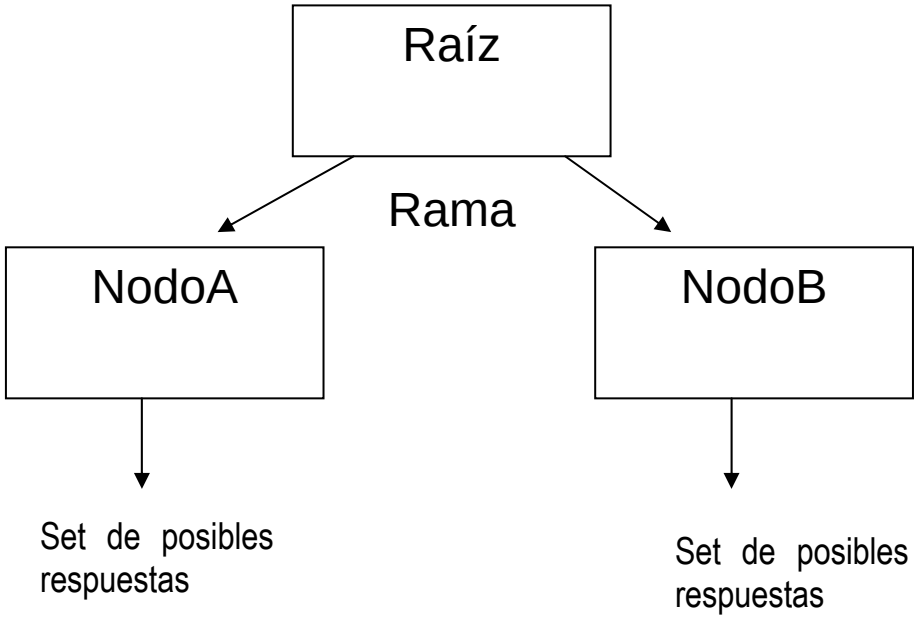
- Ayuda a distinguir entre los diferentes contextos en los que se usa un nodo de decisión, debido a que cada camino de la raíz a una hoja se traduce en una regla distinta.
- Deja de existir la distinción entre nodos que están cerca de la raíz y los que están lejos. Así no hay problemas para reorganizar el árbol si se poda un nodo intermedio.
- Mejora la legibilidad. Las reglas suelen ser más fáciles de entender.

ESTRUCTURAS UTILIZADAS EN EL ALGORITMO C4.5

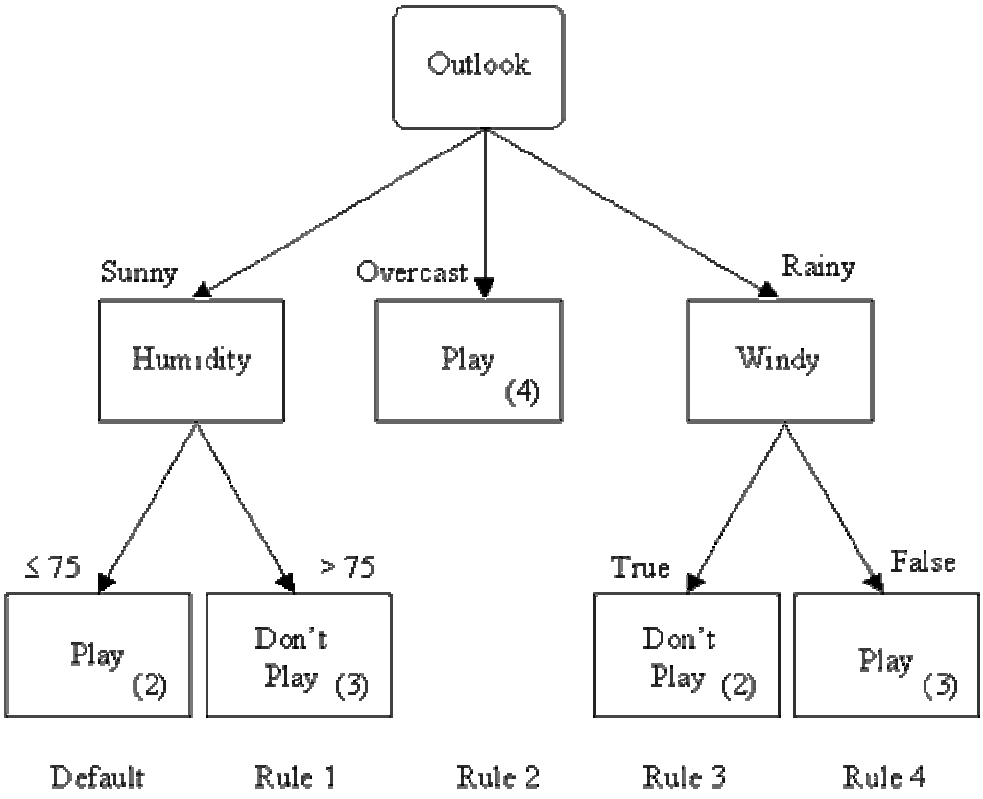
El C4.5 forma parte de la familia de los TDIDT (Top Down Induction Trees), junto con antecesor el ID3.
El C4.5 se basa en el ID3, por lo tanto, la estructura principal de ambos métodos es la misma.
El C4.5 construye un árbol de decisión mediante el algoritmo "divide y vencerás" y evalúa la información en cada caso utilizando los criterios de

Entropía, Ganancia o proporción de ganancia, según sea el caso.

Por otra parte, los árboles de decisión pueden entenderse como una representación de los procesos involucrados en las tareas de clasificación.
Están formados por:
♦ Nodos: Nombres o identificadores de los atributos.
♦ Ramas: Posibles valores del atributo asociado al nodo.
♦ Hojas: Conjuntos ya clasificados de ejemplos y etiquetados con el nombre de una clase.



Ejemplo aplicado de Árbol de Decisión adaptado para **C4.5**



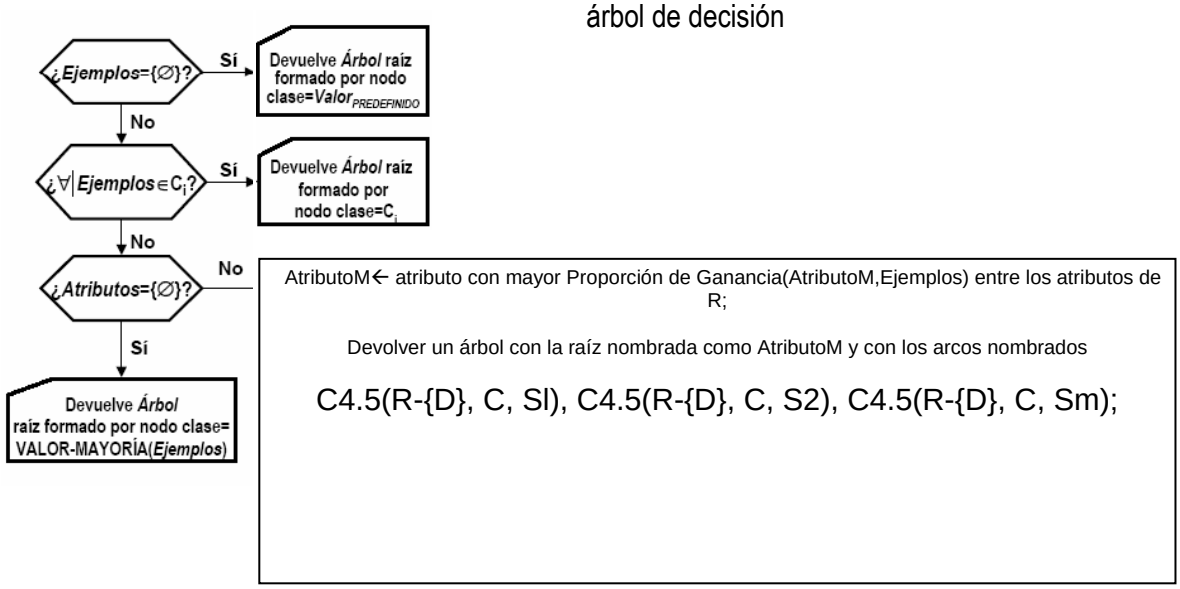
Outlook	Temperature	Humidity	Windy	Play (positive) / Don't Play (negative)
sunny	85	85	false	Don't Play
sunny	80	90	true	Don't Play
overcast	83	78	false	Play
rain	70	96	false	Play
rain	68	80	false	Play
rain	65	70	true	Don't Play
overcast	64	65	true	Play
sunny	72	95	false	Don't Play
sunny	69	70	false	Play
rain	75	80	false	Play
sunny	75	70	true	Play
overcast	72	90	true	Play
overcast	81	75	false	Play
rain	71	80	true	Don't Play

PSEUDOCODIGO DE C4.5

```
Función C4.5
R: conjunto de atributos no clasificadores,
C: atributo clasificador,
S: conjunto de entrenamiento, devuelve un árbol de decisión
Comienzo
    Si S está vacío,
        Devolver un único nodo con Valor Falla;     'para formar el nodo raiz
    Si todos los registros de S tienen el mismo valor para el atributo
clasificador,
        Devolver un único nodo con dicho valor;   'un unico nodo para todos
    Si R está vacío,
        Devolver un único nodo con el valor más frecuente del atributo
        Clasificador en los registros de S [Nota: habrá errores, es decir,
        Registros que no estarán bien clasificados en este caso];
    Si R no está vacío,
        D ←atributo con mayor Proporción de Ganancia (D,S) entre los
atributos de R;
        Sean {dj | j=1,2,...., m} los valores del atributo D;
        Sean {dj | j=1,2,...., m} los subconjuntos de S correspondientes a los
valores de dj respectivamente;
            Devolver un árbol con la raíz nombrada como D y con los arcos
nombrados d1, d2,....,dm, que van respectivamente a los árboles
            C4.5(R-{D}, C, Sl), C4.5(R-{D}, C, S2), C4.5(R-{D}, C, Sm);
Fin
```

Diagrama genérico de algoritmo c4.5

Atributos: conjunto de atributos no clasificadores,
C: atributo clasificador,
Ejemplos: conjunto de ejemplos, devuelve un árbol de decisión



Estimación de la Proporción de Errores para los Árboles de Decisión

Una vez podados, las hojas de los árboles de decisión generados por el C4.5 tendrán dos números asociados: N y E. N es la cantidad de casos de entrenamiento cubiertos por la hoja, y E es la cantidad de errores predichos si un conjunto de N nuevos casos fuera clasificados por el árbol.
La suma de los errores predichos en las hojas, dividido el número de casos de entrenamiento, es un estimador inmediato del error de un árbol podado sobre nuevos casos.

El C4.5 es una extensión del ID3 que acaba con muchas de sus limitaciones. Por ejemplo, permite trabajar con valores continuos para los atributos, separando los posibles resultados en dos ramas: una para aquellos Ai<=N y otra para Ai>N. Además, los árboles son menos frondosos porque cada hoja no cubre una clase en particular sino una distribución de clases, lo cual los hace menos profundos y menos frondosos. Este algoritmo fue propuesto por Quinlan en 1993. El C4.5 genera un árbol de decisión a partir de los datos mediante particiones realizadas recursivamente, según la estrategia de profundidad-primero (depth-first). Antes de cada partición de datos, el algoritmo considera todas las pruebas posibles que pueden dividir el conjunto de datos y selecciona la prueba que resulta en la mayor ganancia de información o en la mayor proporción de ganancia de información. Para cada atributo discreto, se considera una prueba con n resultados, siendo n el número de valores posibles que puede tomar el atributo. Para cada atributo continuo, se realiza una prueba binaria sobre cada uno de los valores que toma el atributo en los datos.

Construcción de un árbol de decisión utilizando el C4.5

Supongamos que queremos construir un árbol de decisión para los siguientes datos:

Estado	Humedad	Viento	Juego tenis
¿	Alta	Leve	No
Soleado	Alta	Fuerte	No
Nublado	Alta	Leve	Si
Lluvia	Alta	Leve	Si
Lluvia	Normal	Leve	Si
Lluvia	Normal	Fuerte	No
Nublado	Normal	Fuerte	Si
Soleado	Alta	Leve	No
Soleado	Normal	Leve	Si
Lluvia	Normal	Leve	Si
Soleado	Normal	Fuerte	Si
Nublado	Alta	Fuerte	Si
Nublado	Normal	Leve	Si
Lluvia	Alta	Fuerte	Si

En este caso, la distribución de datos para el atributo Estado es:

	Desconocido	Soleado	Nublado	Lluvia
No	1	2	0	1
Si	0	2	4	4
Totales	1	4	4	5

Primero calculamos la entropía del conjunto. Recordemos que, como se explicó en la sección 4.4.2.2, no debemos tener en cuenta los atributos desconocidos. Entonces, trabajamos sobre un total de 13 casos, de los cuales 3 son positivos. Tendremos,

$$H(S) = -\frac{3}{13}\log_2\frac{3}{13} - \frac{10}{13}\log_2\frac{10}{13} = 0.7793 bits$$

Calculamos ahora la entropía que tendrían los conjuntos resultantes de la división de datos según este atributo.

$$H(S, Estado) = \frac{4}{13}\left(-\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}\right) + \frac{4}{13}\left(-\frac{0}{4}\log_2\frac{0}{4} - \frac{4}{4}\log_2\frac{4}{4}\right) + \frac{5}{13}\left(-\frac{1}{5}\log_2\frac{1}{5} - \frac{4}{5}\log_2\frac{4}{5}\right) = 0.58536 bits$$

Ahora calculamos la ganancia resultante de dividir al subconjunto según el atributo Estado, tendremos:

$$Ganancia(S, Estado) = \frac{13}{14}(0.7793 - 0.58536) = 0.180 bits$$

Al calcular al información de la división, debemos tener en cuenta una categoría extra para el valor desconocido para el atributo. La información de la división se calcula como:

$$I\_división(S) = -\frac{4}{14}\times\log_2\left(\frac{4}{14}\right) - \frac{4}{14}\times\log_2\left(\frac{4}{14}\right) - \frac{5}{14}\times\log_2\left(\frac{5}{14}\right) - \frac{1}{14}\times\log_2\left(\frac{1}{14}\right) = 1.835 bits$$

Finalmente, calculamos la proporción de ganancia.

$$proporción\_de\_ganancia(S) = \frac{Ganancia(S)}{I\_división(S)} = 0.098 bits$$

De la misma manera en que calculamos la ganancia y la proporción de ganancia para el caso anterior, calculamos para el atributo Humedad los siguientes valores:

Ganancia=0.0746702 bits Proporción de ganancia =0.0746702 bits

Para el caso del atributo Viento obtenemos los siguientes valores:

Ganancia=0.00597769 bits Proporción de ganancia =0.0060687 bits

Al igual que con el ID3, conviene dividir el conjunto según el atributo Estado tanto si trabajamos con la ganancia como si trabajamos con la proporción de ganancia. Al dividir los 14 casos para continuar con la construcción del árbol, los 13 casos para los que el valor de Estado es conocido, no presentan problemas y se reparten según el valor de Estado. Mientras que el caso en que no se conoce el valor de Estado, se reparte entre los conjuntos que tienen Soleado, Nublado y Lluvia con los pesos 4/13, 4/16 y 5/13 respectivamente.

Tomemos por ejemplo, la división de los datos para el valor Nublado del atributo Estado. Los datos que se tienen en cuenta en este caso son:

Estado	Humedad	Viento	Juego tenis	Peso
¿	Alta	Leve	No	4-13
Nublado	Alta	Leve	Si	1
Nublado	Normal	Fuerte	Si	1
Nublado	Alta	Fuerte	Si	1
Nublado	Normal	Leve	Si	1

La distribución de datos para el atributo Humedad es:

	Desconocido	Alta	Normal
No	0	0.3	0
Si	0	2	2
Totales	0	2.3	2

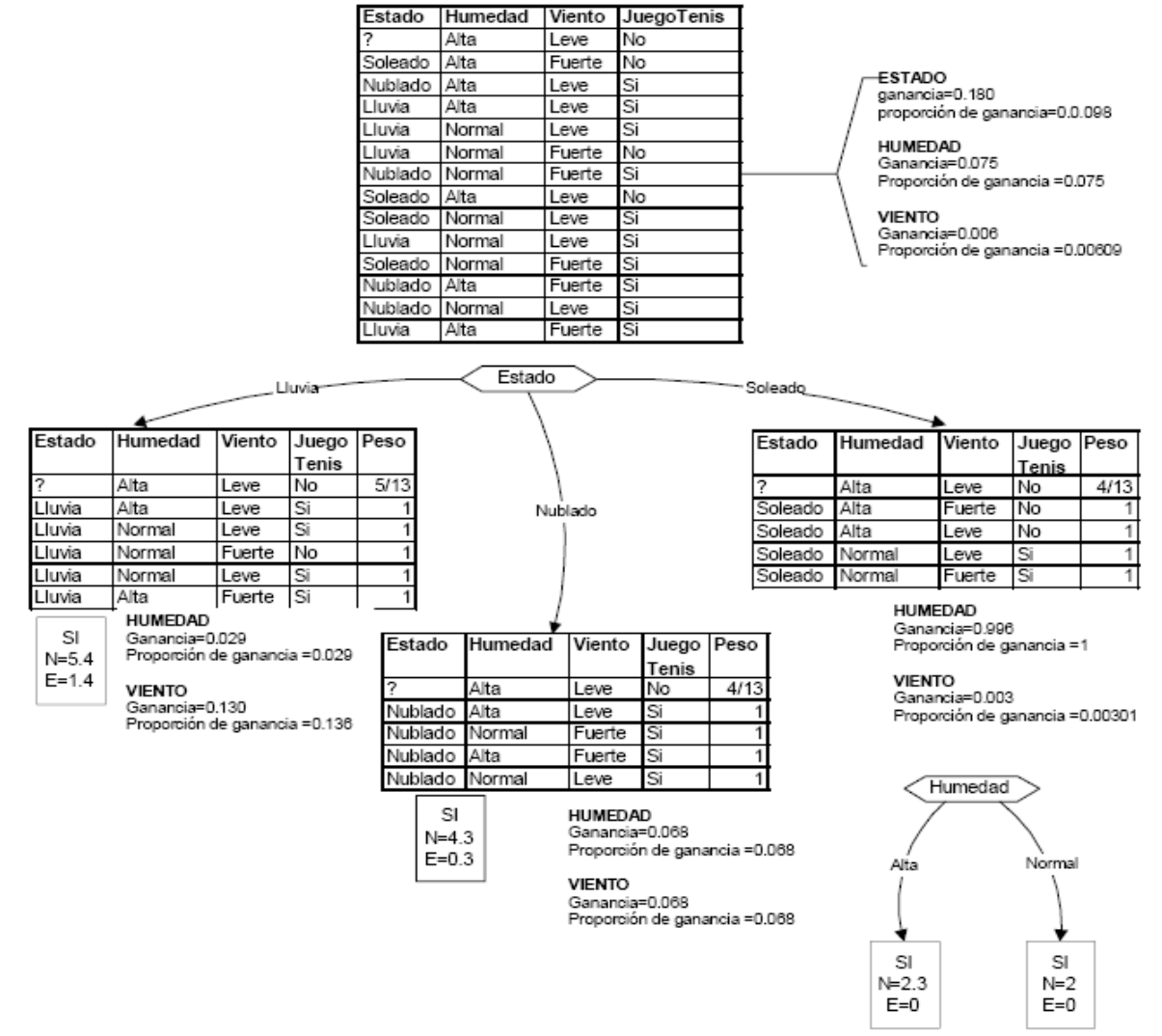
Con estos datos obtenemos para la Humedad los siguientes valores:
Ganancia=0.068 bits Proporción de ganancia =0.068 bits

Para el caso del atributo Viento obtenemos los siguientes valores:
Ganancia=0.068 bits Proporción de ganancia =0.068 bits

En este caso, vemos que la división del conjunto de datos no ofrece ninguna mejora, por lo tanto, colapsamos el árbol a la hoja Si, que es la que mayor peso tiene. La cantidad de casos cubiertos por la hoja, es decir, el N asociado a la misma, es 4.3. Y la cantidad de casos cubiertos incorrectamente, o el E asociado a la hoja, por la hoja son 0.3.

La figura 4.4 muestra un esquema de todos los pasos para la construcción del árbol de decisión en este caso. A continuación se muestra el árbol obtenido.

```
Estado = Nublado: Si (4.3/0.3)
Estado = Lluvia: Si (5.4/1.4)
Estado = Soleado:
Humedad = Alta: No (2.3)
Humedad = Normal: Si (2.0)
```



Estado	Humedad	Viento	JuegoTenis
?	Alta	Leve	No
Soleado	Alta	Fuerte	No
Nublado	Alta	Leve	Si
Lluvia	Alta	Leve	Si
Lluvia	Normal	Leve	Si
Lluvia	Normal	Fuerte	No
Nublado	Normal	Fuerte	Si
Soleado	Alta	Leve	No
Soleado	Normal	Leve	Si
Lluvia	Normal	Leve	Si
Soleado	Normal	Fuerte	Si
Nublado	Alta	Fuerte	Si
Nublado	Normal	Leve	Si
Lluvia	Alta	Fuerte	Si

Estado	Humedad	Viento	Juego Tenis	Peso
?	Alta	Leve	No	5/13
Lluvia	Alta	Leve	Si	1
Lluvia	Normal	Leve	Si	1
Lluvia	Normal	Fuerte	No	1
Lluvia	Normal	Leve	Si	1
Lluvia	Alta	Fuerte	Si	1

Estado	Humedad	Viento	Juego Tenis	Peso
?	Alta	Leve	No	4/13
Soleado	Alta	Fuerte	No	1
Soleado	Alta	Leve	No	1
Soleado	Normal	Leve	Si	1
Soleado	Normal	Fuerte	Si	1

Estado	Humedad	Viento	Juego Tenis	Peso
?	Alta	Leve	No	4/13
Nublado	Alta	Leve	Si	1
Nublado	Normal	Fuerte	Si	1
Nublado	Alta	Fuerte	Si	1
Nublado	Normal	Leve	Si	1

Recordemos que el C4.5 analiza los errores predichos en cada uno de los subárboles y ramas del árbol generado para analizar si es conveniente simplificarlo. En este caso, el error total predicho para el árbol estará dado por:

$$Error\_predicho(Arbol) = 4.3 \times U_{25\%}(0.3,4.3) + 5.4 \times U_{25\%}(1.4,5.4) + 2.3 \times U_{25\%}(0,2.3) + 2 \times U_{25\%}(0,2)$$

Ahora, calculamos el error total predicho de simplificar el árbol por la hoja "Si":

$$Error\_predicho(Arbol\_simplificado) = 14 \times U_{25\%}(4,14) = 5.76$$

El error predicho para el árbol simplificado es menor que el error predicho para el árbol generado. Entonces, el C4.5 poda el árbol a la siguiente hoja:

```
Si (14.0/5.76)
```

Aplicaciones algoritmo C4.5

Simulador Para Volar un Avión Cessna

- Los datos se obtuvieron de 3 pilotos experimentados.
- Haciendo un plan de vuelo asignado 30 veces.
- Cada acción del piloto creaba un ejemplo.
- Se usaron 90,000 ejemplos descritos por 20 atributos.
- Se uso C4.5 que genero un árbol y se convirtió a C.
- Se insertó en el simulador y logro volar.
- Los resultados fueron sorprendentes en el sentido de que aparte de aprender a volar a veces tomaba decisiones mejores que las de sus ``maestros''

Aprendizaje en la WWW

- WebWatcher es un sistema de ayuda de localización de información.
- Aprender las preferencias de un usuario.
- Se registra, cada vez que el usuario selecciona información de la página, o enlaces.
- Esta información está descrita por 530 atributos boléanos indicando si una palabra en particular aparece o no en el texto.

Grúa de embarcación

- Manejar una grúa (6 variables, 450,000 ejemplos)

- Imágenes de alta calidad generadas en tiempo real.
- Gráficos 3D.
- Texturas fotográficas obtenidas en el escenario real de trabajo.
- Información para el usuario y el instructor en pantalla.
- Secuencia de arranque y parada de la grúa real.
- Estiba y desestiba en bodega con grapín.
- Trabajo con varios tipos de materiales, con diferentes densidades y comportamientos.

Sistemas expertos

- Cine.
- Mecánica.
- Autos.
- Diagnostico medico.

Bibliografía

Teoría:
http://www2.cs.uregina.ca/~dbd/cs831/notes/ml/dtrees/c4.5/tutorial.html
http://www.cs.us.es/~delia/sia/html98-99/pag-alumnos/web2/con_teor.html
http://ciberconta.unizar.es/Biblioteca/0007/arboles.html
http://www.cis.temple.edu/~ingargio/cis587/readings/id3-c45.html
http://www.usc.edu/dept/ancntr/Paris-in-LA/Analysis/c45.html
http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t11arbolesslides.pdf

Demos:
http://www2.cs.uregina.ca/~hamilton/courses/831/notes/ml/dtrees/c4.5/tutorial.html