Algoritmos de Inducción (IDE 3)

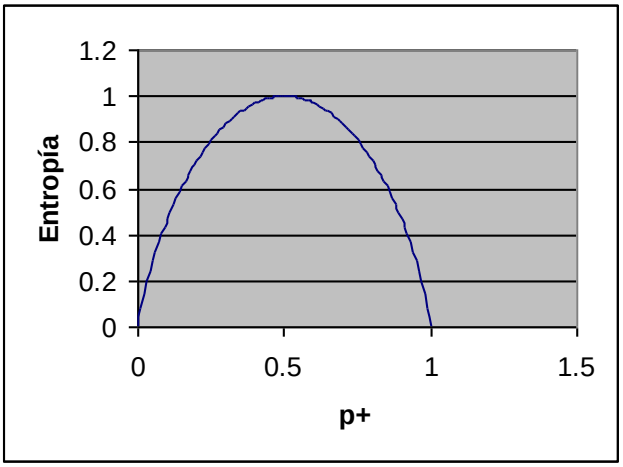
El C4.5 forma parte de la familia de los TDIDT (Top Down Induction Trees), junto con antecesor el ID3. Pertenece a los métodos inductivos del Aprendizaje Automático que aprenden a partir de ejemplos preclasificados. Se utilizan en Minería de datos para modelar las clasificaciones en los datos mediante árboles de decisión. Tanto el ID3 como el C4.5 fueron propuestos por Quinlan, el primero en la década de los ochenta y el segundo en 1993. El C4.5 es una extensión del ID3, que sólo trabaja con valores discretos en los atributos. El C4.5, en cambio, permite trabajar con valores continuos, separando los posibles resultados en dos ramas: una para aquellos Ai<=N y otra para Ai>N. Se genera un árbol de decisión a partir de los datos mediante particiones realizadas recursivamente, aplicando 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.

Estos algoritmos han tenido gran impacto en la Minería de Datos. Forman parte del grupo de sistemas de aprendizaje supervisado. Han tenido muy buena performance en aplicaciones de dominio médico, artificiales y el análisis de juegos de ajedrez. Posee un nivel alto de precisión en la clasificación, pero no hace uso del conocimiento del dominio.

Forma parte de la familia de los Top Down Induction Trees (TDIDT), pertenecientes 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.

Entropía

Es la cantidad de información que se espera observar cuando un evento ocurre según una distribución de probabilidades. **Mide la incertidumbre** dada una distribución de probabilidades. Si tomamos un conjunto con elementos positivos y negativos, la entropía variará entre 0 y 1.



Es 0 si todos los ejemplos pertenecen a la misma clase, y 1 cuando hay igual número de ejemplos positivos y negativos en el conjunto de datos. Cuanto tenemos “C” clases posibles, el valor máximo de la entropía será -> $\mathbf{\log_2 C}$.

La probabilidad de que un ejemplo tomado al azar pertenezca a la clase i y se calcule en base a la frecuencia de los datos de dicha clase en los datos de entrenamiento, se representa en la siguiente fórmula

Poda de los árboles generados

Hay varias razones para podar los árboles generados por los métodos de TDIDT, la sobregeneralización, evaluación de atributos poco importantes o significativos; y el gran tamaño del árbol. Ejemplos con ruido, atributos no relevantes, deben podarse ya que sólo agregan niveles en el árbol y no contribuyen a la ganancia de información. Si el árbol es demasiado grande, se dificulta la interpretación, con lo cual hubiera sido lo mismo utilizar un método de caja negra.

Existen dos enfoques para podar árboles: la pre-poda (preprunning), detiene el crecimiento del árbol cuando la ganancia de información producida al dividir un conjunto no supera un umbral determinado y la post-poda (postprunning), se aplica sobre algunas ramas una vez que se ha terminado.

La pre-poda, no pierde tiempo en construir una estructura que luego será simplificada en el árbol final, busca la mejor manera de partir el subconjunto y evaluar la partición desde el punto de vista estadístico mediante la teoría de la ganancia de información, reducción de errores, etc. Si esta evaluación es menor que un límite predeterminado, la división se descarta y el árbol para el subconjunto es simplemente la hoja más apropiada. Tiene la desventaja de que no es fácil detener un particionamiento en el momento adecuado, un límite muy alto puede terminar con la partición antes de que los beneficios de particiones subsiguientes parezcan evidentes, mientras que un límite demasiado bajo resulta en una simplificación demasiado leve.

La post-poda, es utilizada por el ID3 y el C4.5. Una vez construido el árbol se procede a su simplificación según los criterios propios de cada uno de los algoritmos.

Construcción de un árbol de decisión

A continuación, un ejemplo de aplicación

Estado	Humedad	Viento	Juega Tenis
Soleado	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

A continuación, debemos obtener los siguiente datos, previos a la aplicación del algoritmo:

			SI	10
			NO	4
Estado	Soleado	Nublado	Lluvia	Total
Si	2	4	4	10
No	3	0	1	4
Total	5	4	5	14
Humedad	Alta	Normal	Total	
Si	4	6	10	
No	3	1	4	
Total	7	7	14	
Viento	Leve	Fuerte	Total	
Si	6	4	10	
No	2	2	4	
Total	8	6	14	

1. Calcular entropía del conjunto:

Cuando los casos en un conjunto T contiene ejemplos pertenecientes a distintas clases, se realiza una prueba sobre los distintos atributos y se realiza una partición según el "mejor" atributo. Para encontrar el "mejor" atributo, se utiliza la teoría de la información, que sostiene que la información se maximiza cuando la entropía se minimiza. La entropía determina la azarosidad o desestructuración de un conjunto.

Supongamos que tenemos ejemplos positivos y negativos. En este contexto la entropía de un subconjunto Si, H(Si), puede calcularse como:

$$H(S_i) = -p_i^+ \log p_i^+ - p_i^- \log p_i^-$$

Para nuestro ejemplo se tienen los siguientes datos:

$$H(S) = -p^{Si} \log_2 p^{Si} - p^{No} \log_2 p^{No} = -\frac{10}{14}\log_2 \frac{10}{14} - \frac{4}{14}\log_2 \frac{4}{14} = 0.86312 bits$$

2. Calcular entropía del Subconjunto

Si el atributo at divide el conjunto S en los subconjuntos Si, i = 1,2, ... , n, entonces, la entropía total del sistema de subconjuntos será:

$$H(S, at) = \sum_{i=1}^{n} P(S_i) \cdot H(S_i)$$

Donde $H(S_i)$ es la entropía del subconjunto S_i y $P(S_i)$ es la probabilidad de que un ejemplo pertenezca a S_i. Para nuestro ejemplo si queremos calcular la entropía del atributo estado, se tienen los siguientes datos:

$$H(S, Estado) = \sum_{i=1}^{2} P(S_i) \cdot H(S_i) = \frac{5}{14}\left(-\frac{1}{5}\log_2\frac{1}{5} - \frac{4}{5}\log_2\frac{4}{5}\right) + \frac{4}{14}\left(-\frac{0}{4}\log_2\frac{0}{4} - \frac{4}{4}\log_2\frac{4}{4}\right) + \frac{5}{14}\left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right)$$

$$H(S, Estado) = \frac{5}{14} \times 0.7219 + \frac{4}{14} \times 0 + \frac{5}{14} 0.97095 = 0.6046 bits$$

3. Calcular Ganancia

La ganancia en información puede calcularse como la disminución en entropía, mediante el siguiente cálculo:

$$I(S, at) = H(S) - H(S, at)$$

Donde $H(S)$ es el valor de la entropía a priori, antes de realizar la subdivisión, y $H(S, at)$ es el valor de la entropía del sistema de subconjuntos generados por la partición según at.
Para el ejemplo de Ganancia, se tendría el siguiente:

$$Ganancia(S, Estado) = H(S) - H(S, Estado) = 0.25852 bits$$

4. Calcular Proporción de Ganancia

Favorece a los atributos que tienen muchos valores frente a los que tienen pocos valores. Si se tiene un conjunto de registros con fecha y se particiona según el campo fecha, se obtendrá un árbol perfecto, pero que no servirá para clasificar casos futuros, dato el gran tamaño del mismo. Para resolver esta situación se divide a los datos de entrenamiento en conjuntos pequeños, con lo cual tendrá una alta ganancia de información.
Una alternativa para dividir a los datos es la ganancia de información. Esta medida penaliza a los atributos como fecha al incorporar el término de información de la división.
La información de la división penalizará a aquellos atributos con muchos valores uniformemente distribuidos. Si tenemos n datos separados perfectamente por un atributo, la información de la división para ese caso será log2n. En cambio, un atributo que divide a los ejemplos en dos mitades, tendrá una información de la división de 1.
Cuando la información de la división es cercana a cero, pueden aplicarse varias heurísticas. Puede utilizarse la ganancia como medida y utilizar la proporción de ganancia sólo para los atributos que estén sobre el promedio.

a. Cálculo de I_división

$$I\_división(X) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \times \log_2\left(\frac{|T_i|}{|T|}\right)$$

Para el atributo estado, el cálculo es:

$$I\_división(S) = -\sum_{i=1}^{n} \frac{|S_i|}{|S|} \times \log_2\left(\frac{|S_i|}{|S|}\right) = -\frac{5}{14} \times \log_2\left(\frac{5}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{5}{14} \times \log_2\left(\frac{5}{14}\right) = 1.577 bits$$

b. Cálculo de Proporción de Ganancia

$$proporción\_de\_ganancia(X) = \frac{I(T,X)}{I\_división(X)}$$

Para el atributo Estado, el cálculo es:

$$proporción\_de\_ganancia(S) = \frac{Ganancia(S)}{I\_división(S)} = 0.491042 bits$$

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:

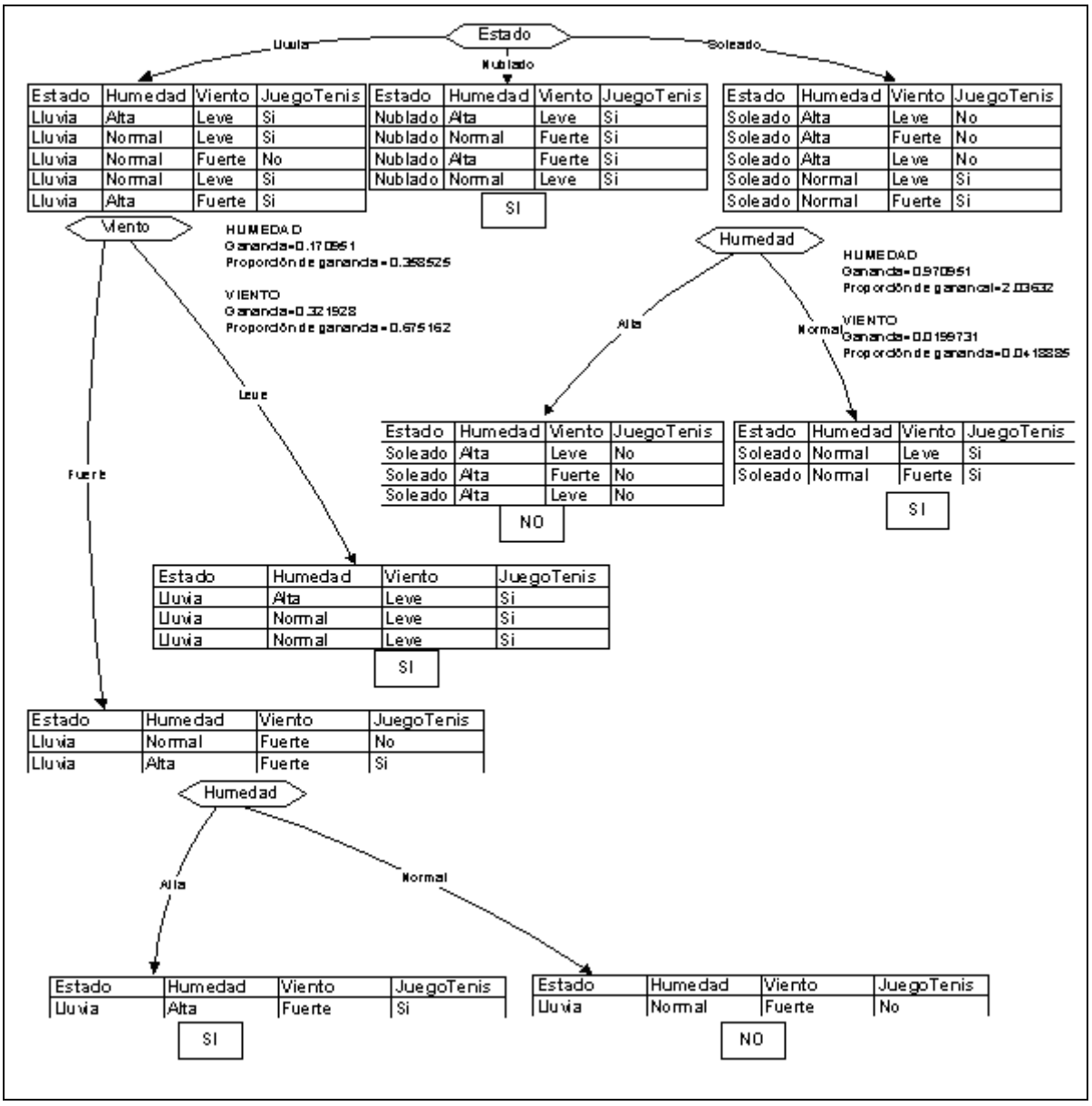
- Ganancia=0.0746702 bits
- Proporción de ganancia =0.14934 bits

Para el caso del atributo **Viento** obtenemos los siguientes valores:

- Ganancia=0.00597769 bits
- Proporción de ganancia =0.0122457 bits

	Estado	Humedad	Viento
Ganancia	0.258	0.074	0.005
Proporción Ganancia	0.491	0.149	0.012

Una vez que hemos calculado las ganancias y proporciones de ganancia para todos los atributos disponibles, debemos elegir el atributo según el cual dividiremos a este conjunto de datos. Recordemos que tanto en el caso de la ganancia como en el de la proporción de ganancia, el mejor atributo para la división es aquel que la maximiza.
En este ejemplo, la división según el atributo Estado es la que mayor ganancia y proporción de ganancia ofrece. Esto significa que el nodo raíz del árbol será un nodo que evalúa el atributo Estado.
Esquema de la construcción de un árbol de decisión utilizando el ID3



Generación de reglas.

Tomemos otro ejemplo, donde se puede observar los resultados de una regla para un set de datos y la interpretación de los resultados. Suponga que hay 140 observaciones en los datos de entrenamiento, de los cuales 50 pertenecen a la clase de la Verginica. Además, se sabe que la regla arriba descripta se aplica sobre 45 registros, de los cuales 44 son verginica.
Los resultados son los siguientes:

SOPORTE: 32% (= 45 / 140 * 100)
CONFIANZA: 98% (= 44 / 45 * 100)
CAPTURA: 88 % (= 44 / 50 * 100)

Clases	Setosa	Verginica	Versicolor	Total
OBS	50	50	40	140

Regla 2	0	44	1	45
	Setosa	Verginica	Versicolor	Total
	Resultados	Valores	Fórmulas	
	Confianza	97,78	44/45*100	
	Captura	88	44/50*100	
	Soporte	32,14	45/140*100	

La **confianza**, esta dada por la relación que existe entre la totalidad de las observaciones que fueron afectados por la regla (45) y la cantidad de observaciones que fueron afectadas por la clase mayoritaria (44) con esta misma regla. Para este caso la confianza es del 97,78 %.

El porcentaje de **captura**, esta dado por la relación que existe entre la cantidad de observaciones de la clase mayoritaria que fueron afectadas por esta regla (44) y la cantidad total de observaciones procesadas pertenecientes a esta misma clase (50). Para este caso el resultado es del 88%.

La cantidad de observaciones que **soporta** la regla, esta dada por la relación que existe entre la totalidad de las observaciones que afecta la regla (45) y la totalidad de observaciones procesadas (140). El resultado es del 32, 14%.

Criterios de ramificación

A medida que el árbol se va desarrollando, termine o no de ramificarse un nodo y se declare al nodo como un nodo hoja, puede ser determinado por los siguientes criterios. Se puede no elegir ningún criterio, uno o varios. Si no se elige ningún criterio, la aplicación usa los valores por defecto.

- Tamaño mínimo del nodo (Valor por defecto = 5 registros)
  El nodo no se ramifica más si el número de registros en el nodo es = (porcentaje a ingresar) o menor al número total de registros.
- Nivel máximo de pureza (Valor por defecto = 100% de pureza)
  El nodo no se ramifica más si el valor de pureza es = (porcentaje a ingresar) o mayor. (Por ejemplo si la pureza es del 90 % significa que ese porcentaje de registros en el nodo pertenecen en un 90 % a la clase mayoritaria)
- Nivel máximo de profundidad (Valor por defecto = 20 es el máximo nivel de profundidad)
  El nodo no se ramifica más si el valor de la profundidad es = (valor a ingresar) o mayor. (El nodo raíz tiene profundidad 1. Cualquier nodo dependiente es igual a la profundidad de su nodo padre + 1).