Arboles de Decisión (II)

Carlos Hurtado L.
Depto de Ciencias de la Computación,
Universidad de Chile

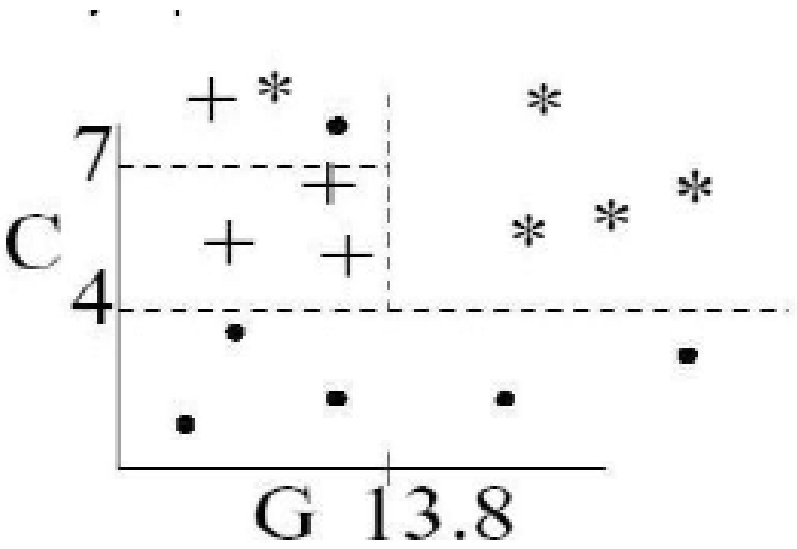
Contenido

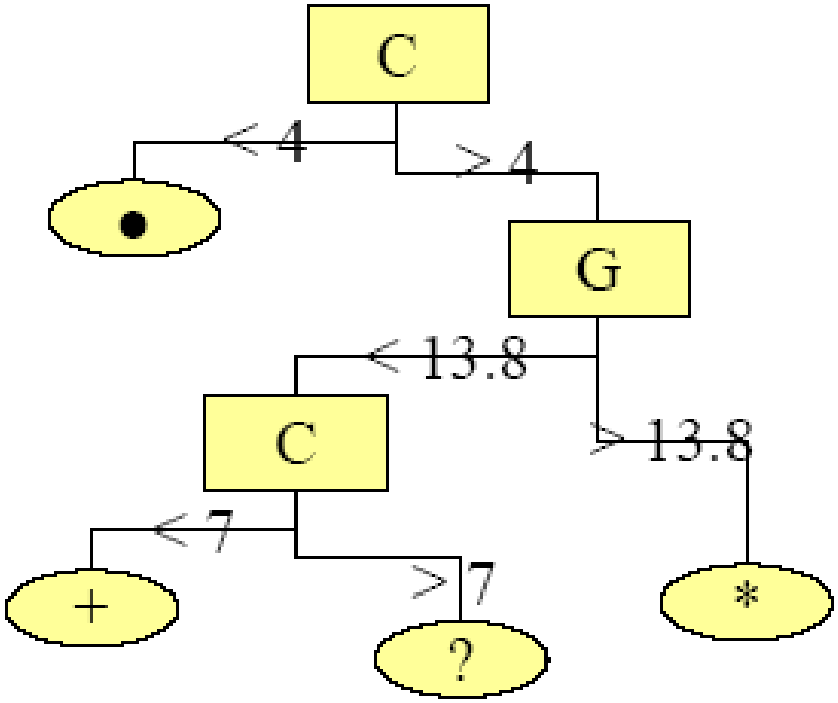
- Repaso algoritmo de Hunt
- Selección de Splits usando índice Gini
- Selección de Splits usando entropía
- Medidas de Impureza
- Selección de Splits usando test Chi-cuadrado
- Medidas de Separación

Contenido

- Repaso algoritmo de Hunt
- Selección de Splits usando índice Gini
- Selección de Splits usando entropía
- Selección de Splits usando test Chi-cuadrado
- Medidas de Separación

Arboles de Decisión

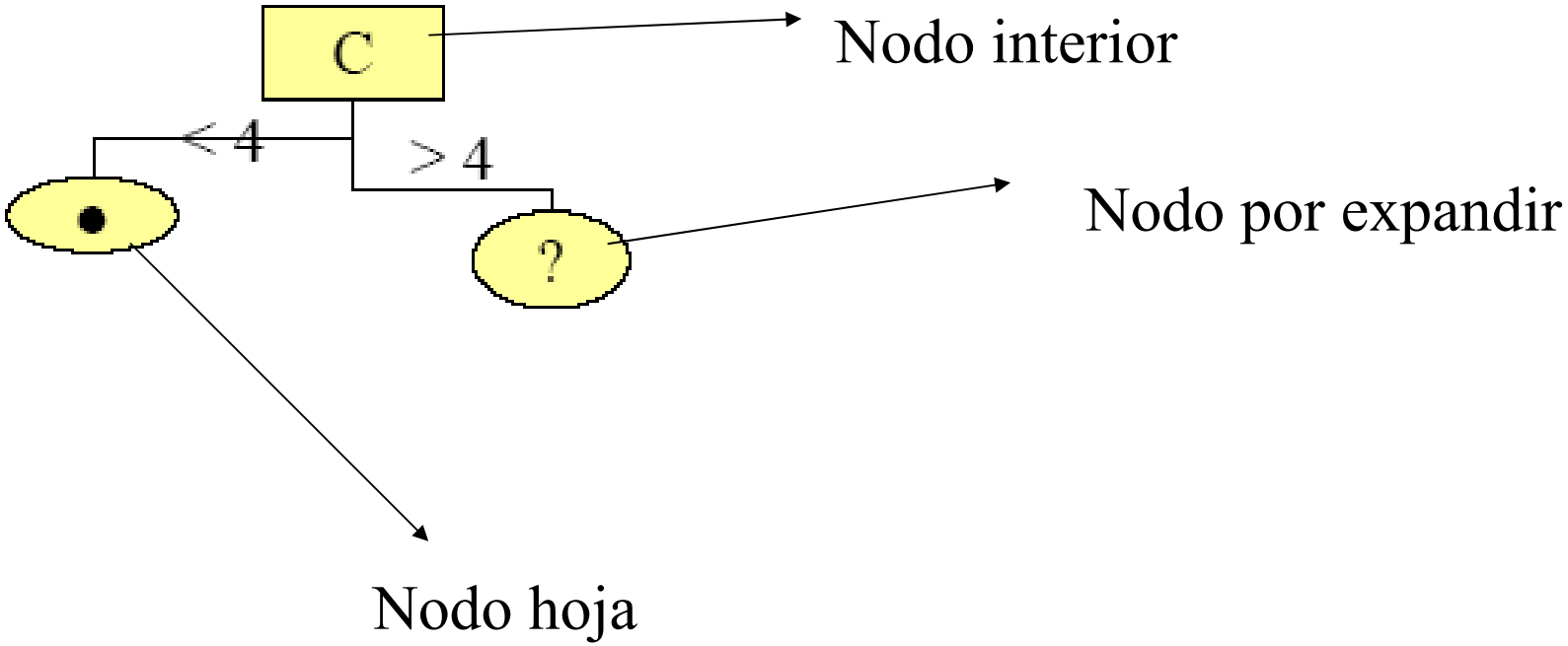




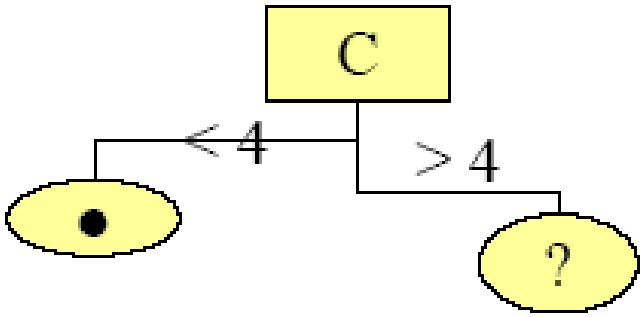
Algoritmo de Hunt

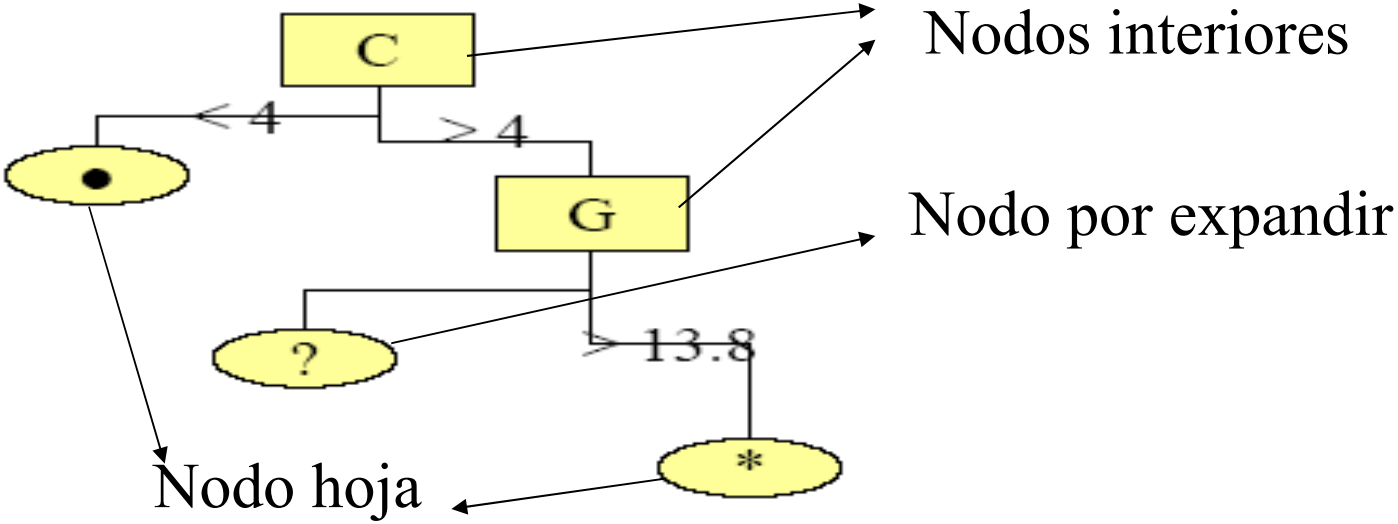
- Idea básica: cada nodo en el árbol de decisión tiene asociado un subconjunto de los datos de entrenamiento
- Inicialmente, el nodo raíz tiene asociado todo el conjunto de entrenamiento
- Construimos un árbol parcial que tiene tres tipos de nodos:
  - Expandidos (interiores)
  - Hojas: serán hojas en el árbol final y tienen asociada una clase
  - Nodos por expandir: son hojas en el árbol parcial, pero deben ser expandidos
- Operación de expansión de un nodo t:
  - Encontrar el mejor split para t
  - Particionar los datos de t en nodos hijos de acuerdo al split
  - Etiquetar t y sus nodos hijos con el mejor split

Algoritmo de Hunt (I)

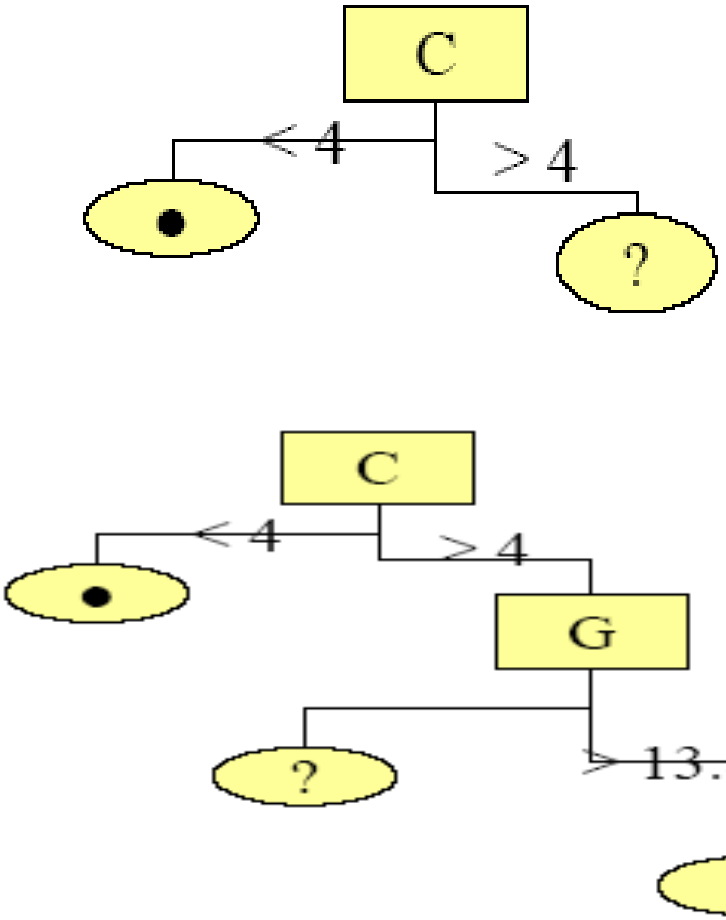


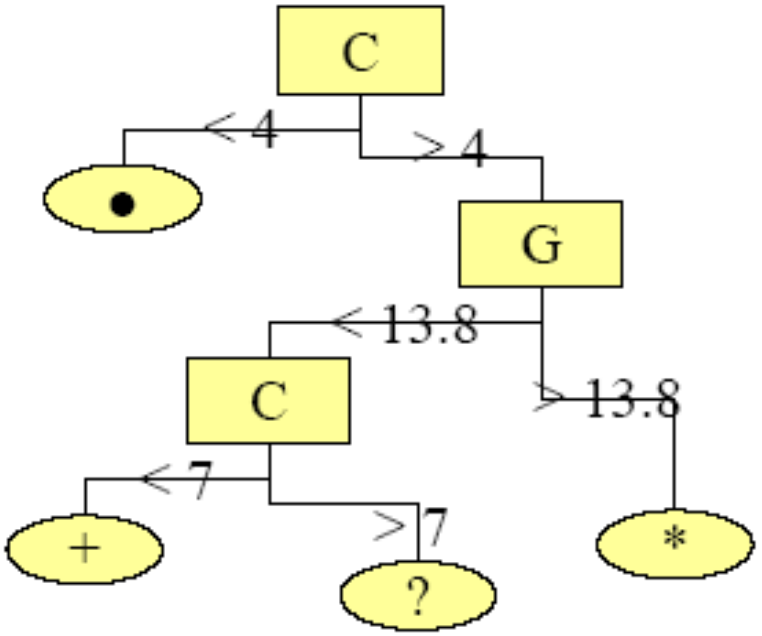
Algoritmo de Hunt (II)





Algoritmo de Hunt (III)





Algoritmo de Hunt

- Main(Conjunto de Datos T)
  - Expandir(T)
- Expandir(Conjunto de Datos S)
  - If (todos los datos están en la misma clase) then return
  - Encontrar el mejor split r
  - Usar r para particonar S en S1 y S2
  - Expandir(S1)
  - Expandir(S2)

Algoritmo de Hunt: observaciones

- Estrategia recursiva y "greedy"
- Las expansiones se realizan "primero en profundidad"
- Lo complejo es encontrar el mejor split en cada expansión
- Número de splits candidatos depende del tipo de atributo (categórico o numérico) y del tipo de split (e.g., complejo vs. simple).

¿Cuál es el mejor split?

- Buscamos splits que generen nodos hijos con la menor impureza posible (mayor pureza posible)
- Existen distintos métodos para evaluar splits. **Criterios de Selección**:
  - Indice Gini
  - Entropía (Ganancia de información)
  - Test Chi-cuadrado
  - Proporción de Ganancia de Información

Contenido

- Repaso algoritmo de Hunt
- Selección de Splits usando índice Gini
- Selección de Splits usando entropía
- Medidas de Impureza
- Selección de Splits usando test Chi-cuadrado
- Medidas de Separación

Selección de splits usando índice Gini

- Recordemos que cada nodo del árbol define un subconjunto de los datos de entrenamientos
- Dado un nodo t del árbol, $Gini(t)$ mide el grado de impureza de t con respecto a las clases
  - Mayor $Gini(t)$ implica mayor impureza
  - $Gini(t)$ = 1 – Prob. De sacar dos registros de la misma clase

Indice Gini

- Recordar que el nodo t tiene asociado un subconjunto de los datos

- $Gini(t)$: probabilidad de NO sacar dos registros de la misma clase del nodo t

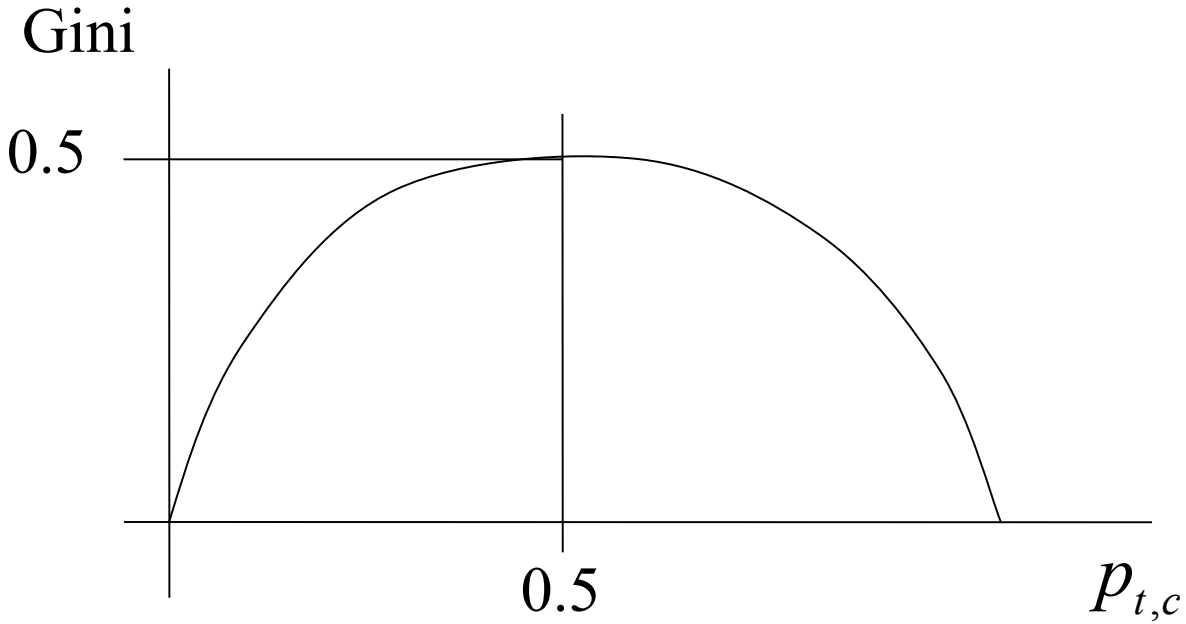
$$Gini(t) = 1 - \sum_{c \in C} p_{t,c}^{\ 2}$$

donde C es el conjunto de clases y $p_{t,c}$ es la prob. de ocurrencia de la clase c en el nodo t

Indice Gini: ejemplo

C_1	C_2	Gini
0	6	0
1	5	0.278
2	4	0.444
3	3	0.5

Indice Gini



Selección de Splits: GiniSplit

- Criterio para elegir un split: selecciónar el split con menor gini ponderado (GiniSplit)
- Dado un split S={s1,…,sn} de t

$$GiniSplit(t,s) = \sum_{s \in S} \frac{|s|}{|t|} Gini(s)$$

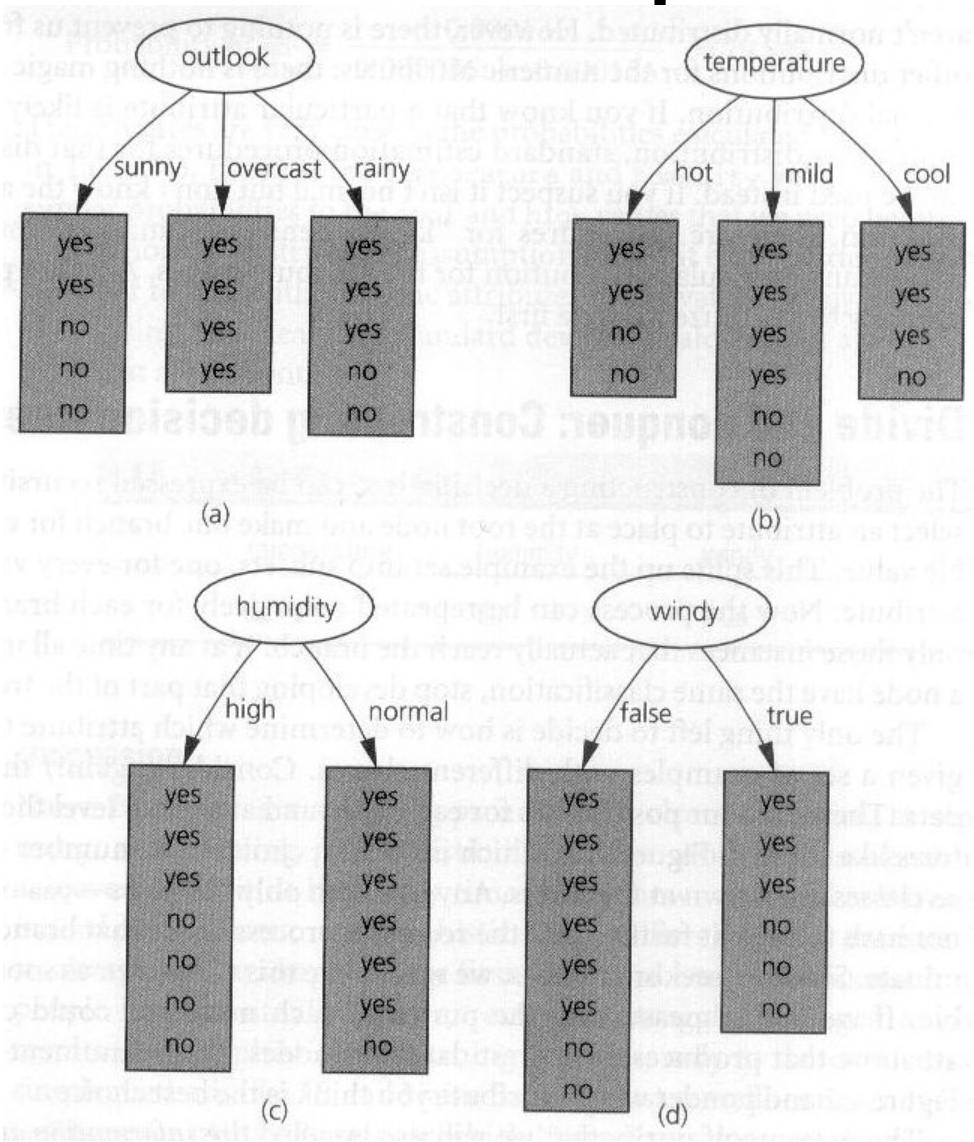
Ejemplo: weather.nominal

Outlook	Temp.	Humidity	Windy	Play
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Rainy	Mild	High	FALSE	Yes
Rainy	Cool	Normal	FALSE	Yes
Rainy	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Sunny	Mild	High	FALSE	No
Sunny	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	FALSE	Yes
Sunny	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Mild	High	TRUE	No

Weather.nominal: splits

- En este caso todos los atributos son nominales
- En este ejemplo usaremos *splits* simples

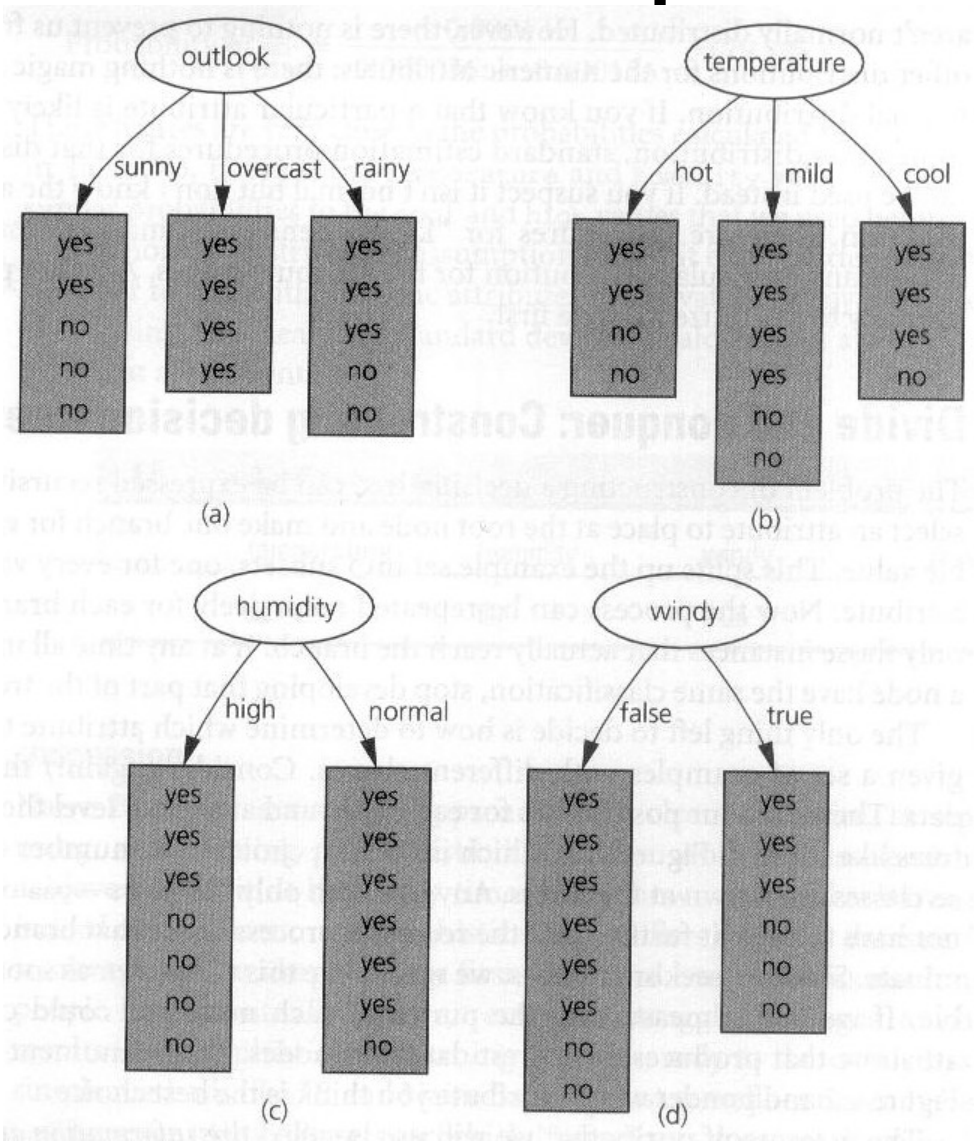
Posibles splits



weather.nominal: atributos vs. clases

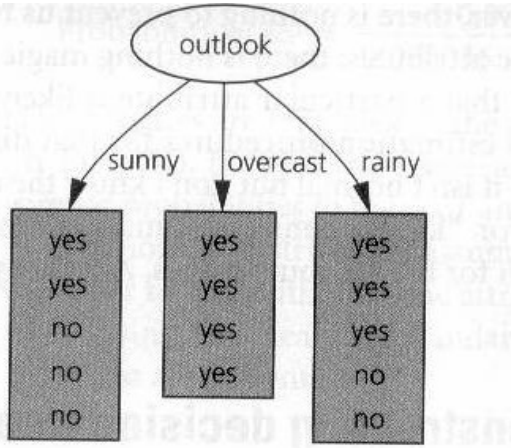
Outlook			*Temperature*			*Humidity*			*Windy*			*Play*	
	Yes	*No*		*Yes*	*No*		*Yes*	*No*		*Yes*	*No*	*Yes*	*No*
Sunny	2	3	Hot	2	2	High	3	4	FALSE	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	TRUE	3	3		
Rainy	3	2	Cool	3	1								

Posibles splits



Ejemplo

- Split simple sobre variable Outlook
  - Gini(sunny) = 1- 0.16 - 0.36 = 0.48
  - Gini(overcast)= 0
  - Gini(rainy)= 0.48
  - GiniSplit = (5/14) 0.48 + 0 0.48 + (5/14) 0.48 = 0.35



Ejercicio: selección de splits usando Gini para weather.nominal

1. Dar el número de splits que necesitamos evaluar en la primera iteración del algoritmo de Hunt, para (a) splits complejos y (b) splits simples.
2. Seleccionar el mejor split usando el criterio del índice Gini
   - Calcular el GiniSplit de cada split
   - Determinar el mejor split

Contenido

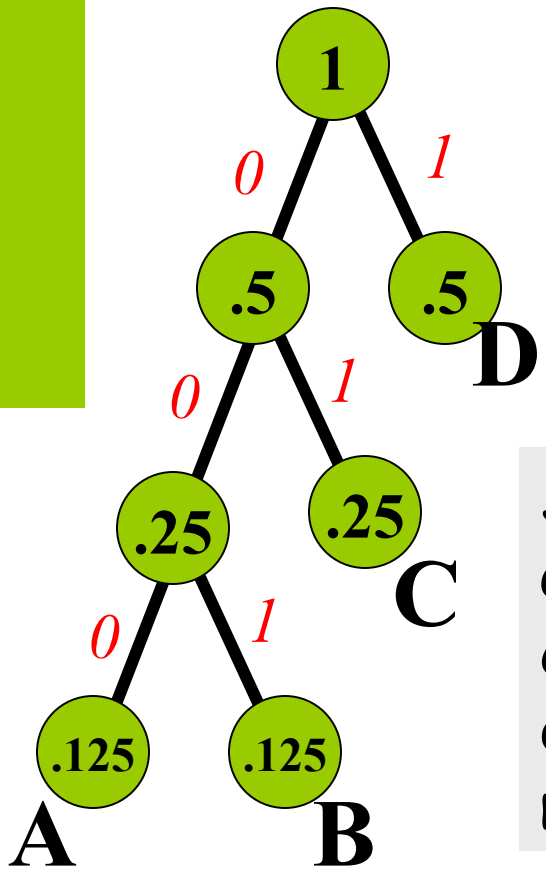
- Repaso algoritmo de Hunt
- Selección de Splits usando índice Gini
- Selección de Splits usando entropía
- Medidas de Impureza
- Selección de Splits usando test Chi-cuadrado
- Medidas de Separación

Teoría de Información

- Supongamos que trasmitimos en un canal (o codificamos) mensajes que son símbolos en {x1,x2,…,xn}.
- A primera vista, necesitamos log n bits por mensaje.
- Si conocemos la distribución de probabilidades P(X=xi)=pi, podríamos necesitar menos bits por mensaje.

Códigos de Huffman

Símb.	Prob.
A	.125
B	.125
C	.25
D	.5



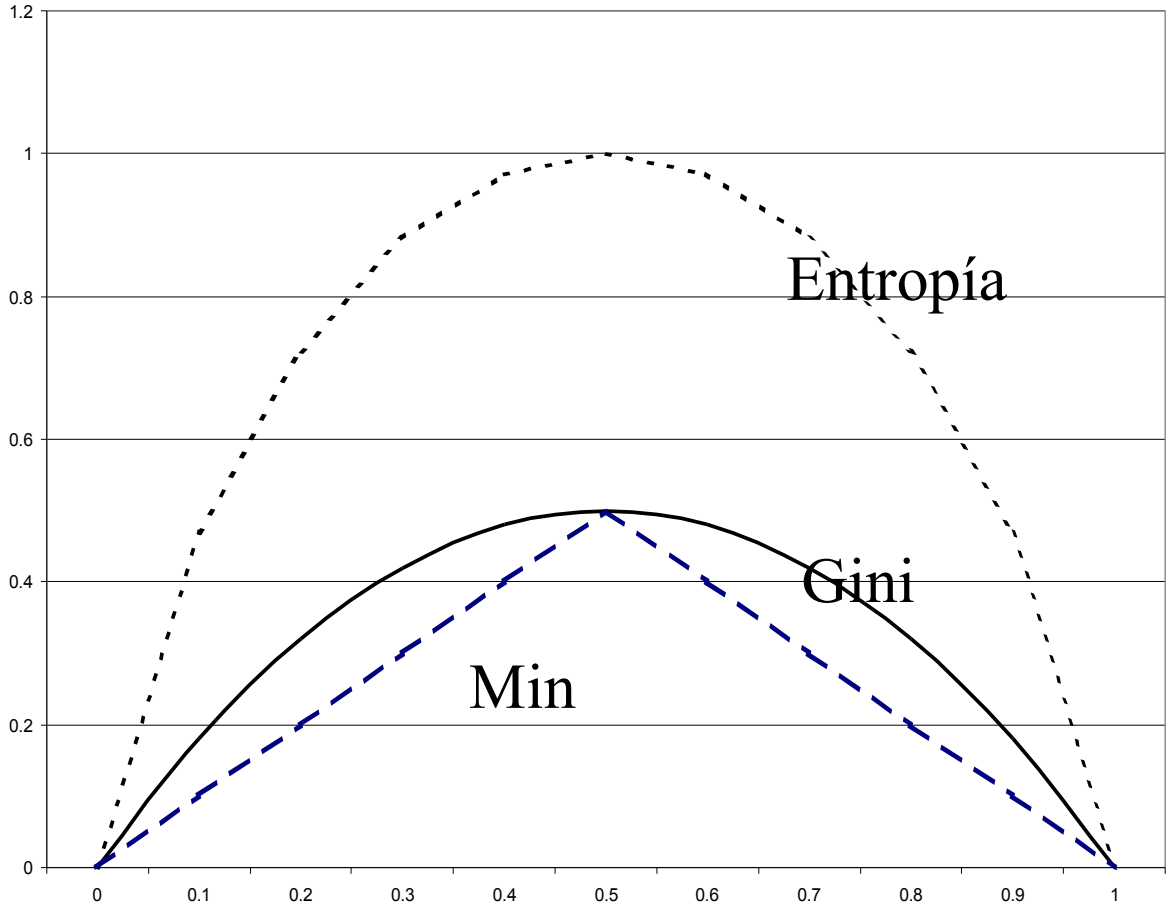
M	cod.	long.	prob	
A	000	3	0,125	0,375
B	001	3	0,125	0,375
C	01	2	0,250	0,500
D	1	1	0,500	0,500
average message length				1,750

Si usamos este código para enviar mensajes (A,B,C, o D) con la distribución de prob. dada, en promedio cada mensaje require 1.75 bits.

Entropía

- Entropía: mínimo teórico de bits promedio necesarios para trasmitir un conjunto de mensajes sobre {x1,x2,…,xn} con distribución de prob. P(X=xi)=pi
- Entropy(P) = $-(p_1 * \log(p_1) + p_2 * \log(p_2) + .. + p_n * \log(p_n))$
- Información asociada a la distribución de probabilidades P.
- Ejemplos:
  - Si P es (0.5, 0.5), Entropy(P) = 1
  - Si P es (0.67, 0.33), Entropy(P) = 0.92
  - Si P is (1, 0), Entropy(P)=0
- Mientras más uniforme es P, mayor es su entropía

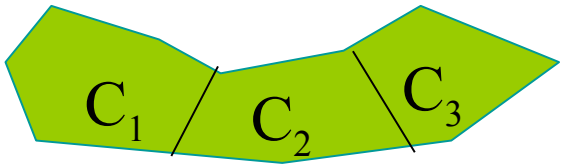
Entropía



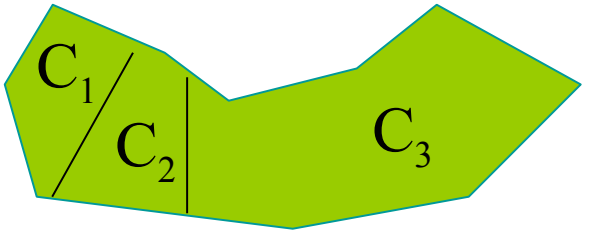
Selección de splits usando Entropía

$$Entropy(t) = \sum_{c \in C} - p_{t,c} \log_2 p_{t,c}$$

- La entropía mide la impureza de los datos S
- Mide la información (num de bits) promedio necesaria para codificar las clases de los datos en el nodo t
- Casos extremos
  - Todos los datos pertenecen a la misma clase (Nota: 0·log 0 = 0)
  - Todas las clases son igualmente frecuentes



Entropía alta



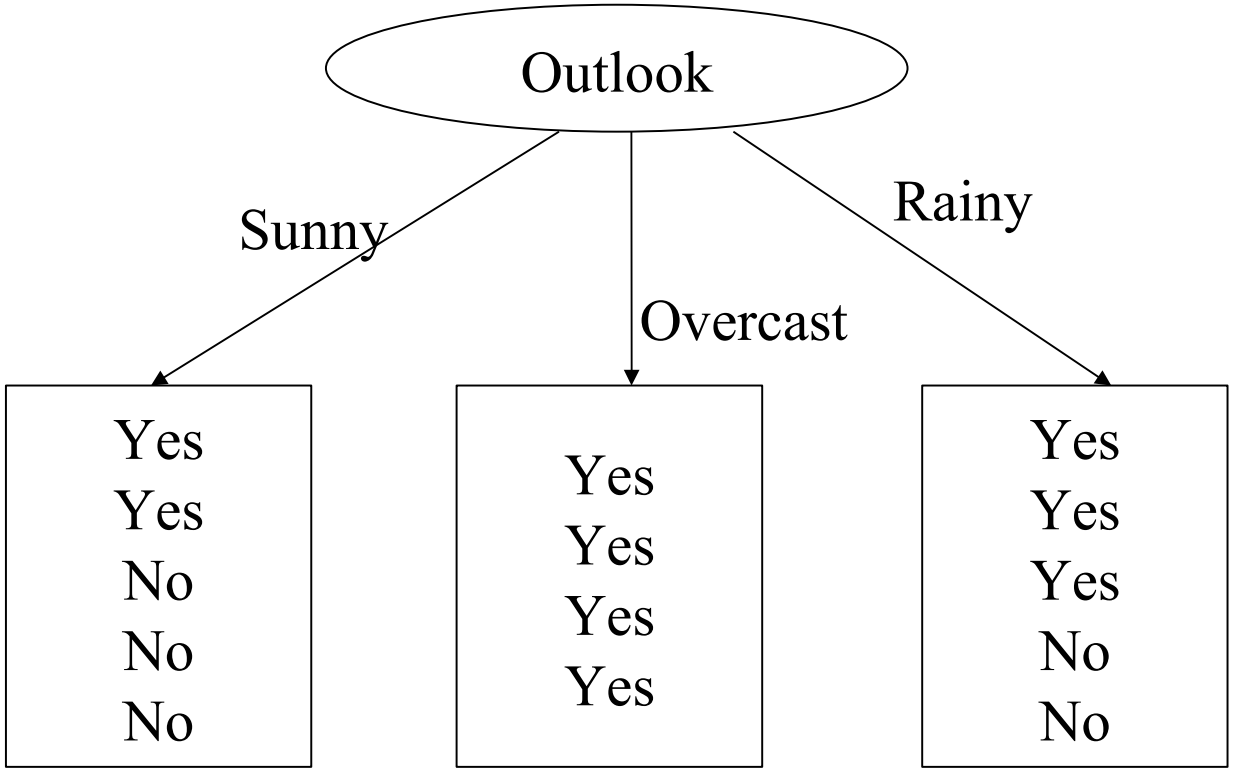
Entropía baja

Selección de Splits: Ganancia de Información

- Criterio para elegir un split: selecciónar el split con la mayor ganancia de información (Gain)
- Equivalente a seleccionar el split con la menor entropía ponderada
- Dado un split S={s1,…,sn} de t

$$Gain(t,S) = Entropy(t) - \sum_{s \in S} \frac{|s|}{|t|} Entropy(s)$$

Ejemplo: supongamos que elegimos el primer nodo



Ejemplo: Cálculo de la Entropía para split simpla basado en Outlook

$$Entropy(S) = \sum_{i=1}^{2} - p_i \log_2 p_i$$

$$= -\frac{9}{14} \cdot \log_2 \frac{9}{14} - \frac{5}{14} \cdot \log_2 \frac{5}{14} = 0.92$$

$$Entropy(S_{sunny}) = -\frac{2}{5} \cdot \log_2 \frac{2}{5} - \frac{3}{5} \cdot \log_2 \frac{3}{5} = 0.97$$

$$Entropy(S_{overcast}) = -\frac{4}{4} \cdot \log_2 \frac{4}{4} - \frac{0}{4} \cdot \log_2 \frac{0}{4} = 0.00$$

$$Entropy(S_{rainy}) = -\frac{3}{5} \cdot \log_2 \frac{3}{5} - \frac{2}{5} \cdot \log_2 \frac{2}{5} = 0.97$$

Ejemplo: Cálculo de la Ganancia de información

$$Gain(S, outlook) = Entropy(S) - \sum_{v \in Values(a)} \frac{|S_v|}{|S|} Entropy(S_v)$$

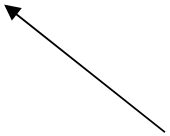
$$= 0.92 - \frac{5}{14} \cdot 0.97 - \frac{4}{14} \cdot 0 - \frac{5}{14} \cdot 0.97$$

$$= 0.92 - 0.69 = 0.23$$

$$Gain(S, temp) = 0.03$$

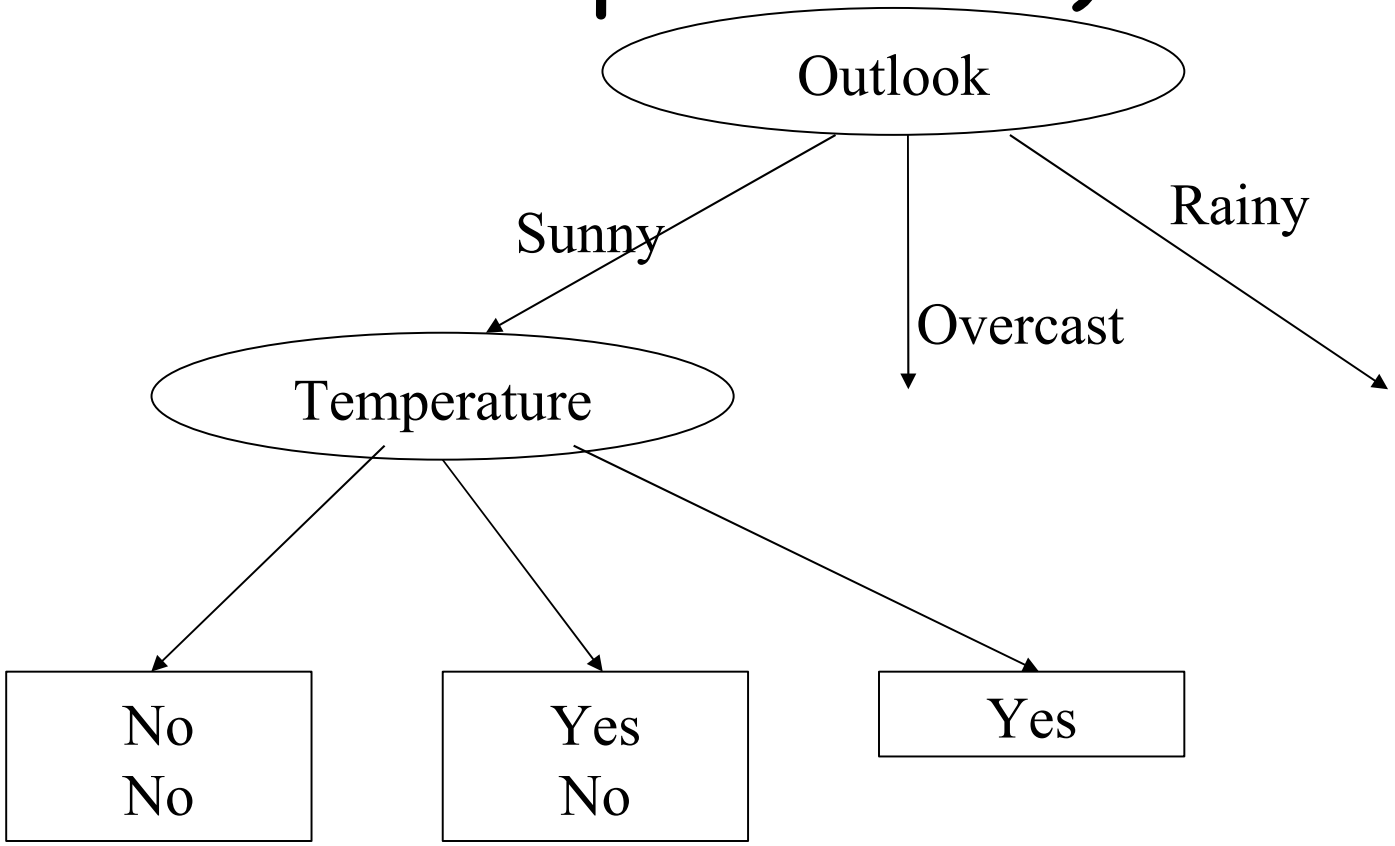
$$Gain(S, humidity) = 0.15$$

$$Gain(S, windy) = 0.05$$



Seleccionar!

Ejercicio: Expansión de nodo "Sunny" (probemos el split "Temperature")



Cálculo de la Entropía para los nodos del split "Temperature"

$$Entropy(S) = 0.97$$

$$Entropy(S_{hot}) = -\frac{0}{2} \cdot \log_2 \frac{0}{2} - \frac{2}{2} \cdot \log_2 \frac{2}{2} = 0$$

$$Entropy(S_{mild}) = -\frac{1}{2} \cdot \log_2 \frac{1}{2} - \frac{1}{2} \cdot \log_2 \frac{1}{2} = 1$$

$$Entropy(S_{cool}) = -\frac{1}{1} \cdot \log_2 \frac{1}{1} - \frac{0}{1} \cdot \log_2 \frac{0}{1} = 0$$

Cálculo de la Ganancia para "Temperature"

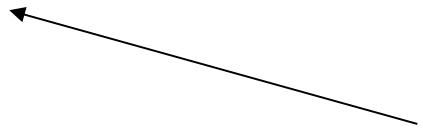
$$Gain(S, temp) = Entropy(S) - \sum_{v \in Values(a)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$= 0.97 - \frac{2}{5} \cdot 0 - \frac{2}{5} \cdot 1 - \frac{1}{5} \cdot 0$$

$$= 0.97 - 0.40 = 0.57$$

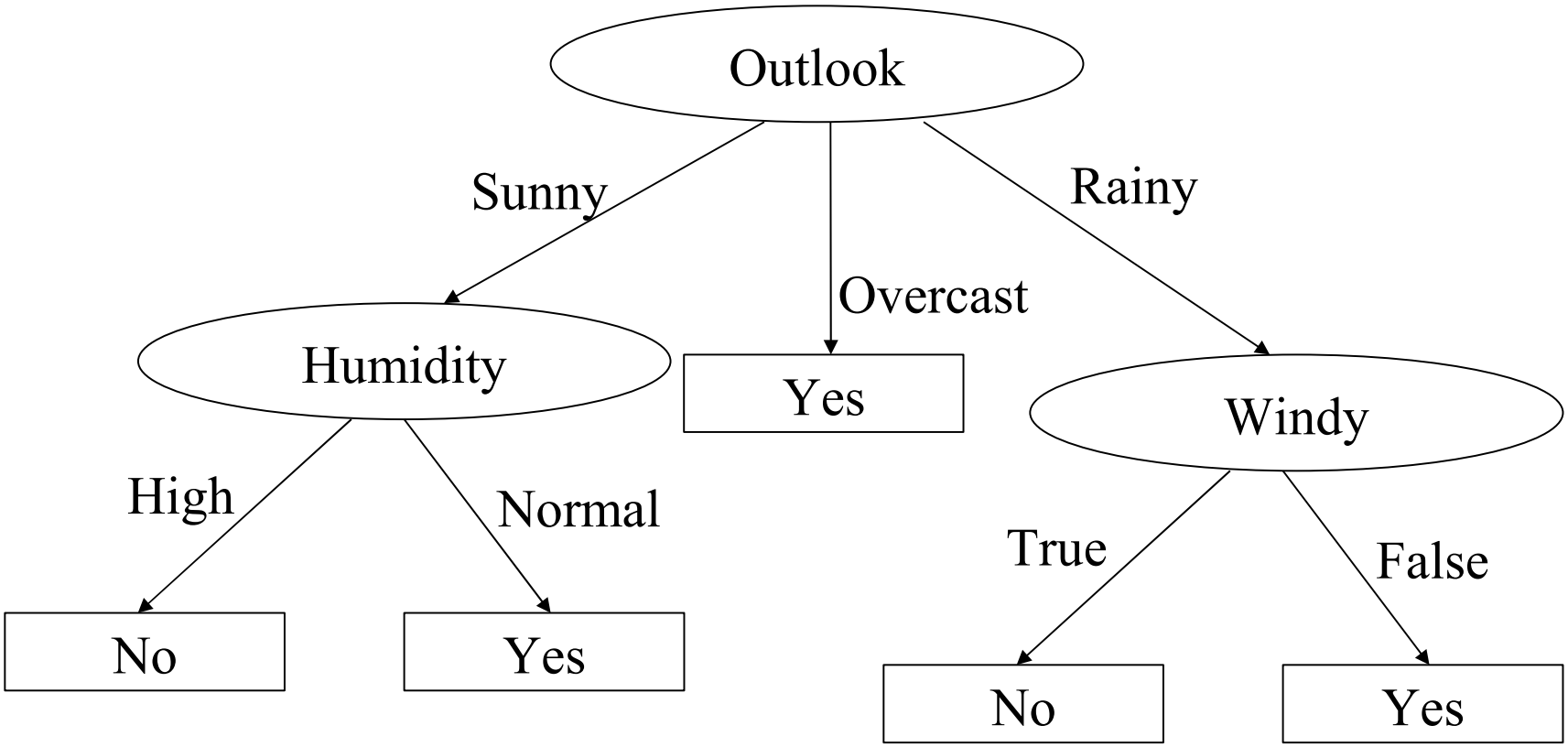
$$Gain(S, humidity) = 0.97$$

$$Gain(S, windy) = 0.02$$



La ganancia de "humidity" es mayor
Por lo que seleccionamos este split

Ejemplo: Arbol resultante



Contenido

- Repaso algoritmo de Hunt
- Selección de Splits usando índice Gini
- Selección de Splits usando entropía
- Medidas de Impureza
- Selección de Splits usando test Chi-cuadrado
- Medidas de Separación

Medidas de Impureza

$\Delta_k = \{x \in \mathbb{R}^k : x \geq 0, \sum_i x_i = 1\}$ es el simplex de dimensión k

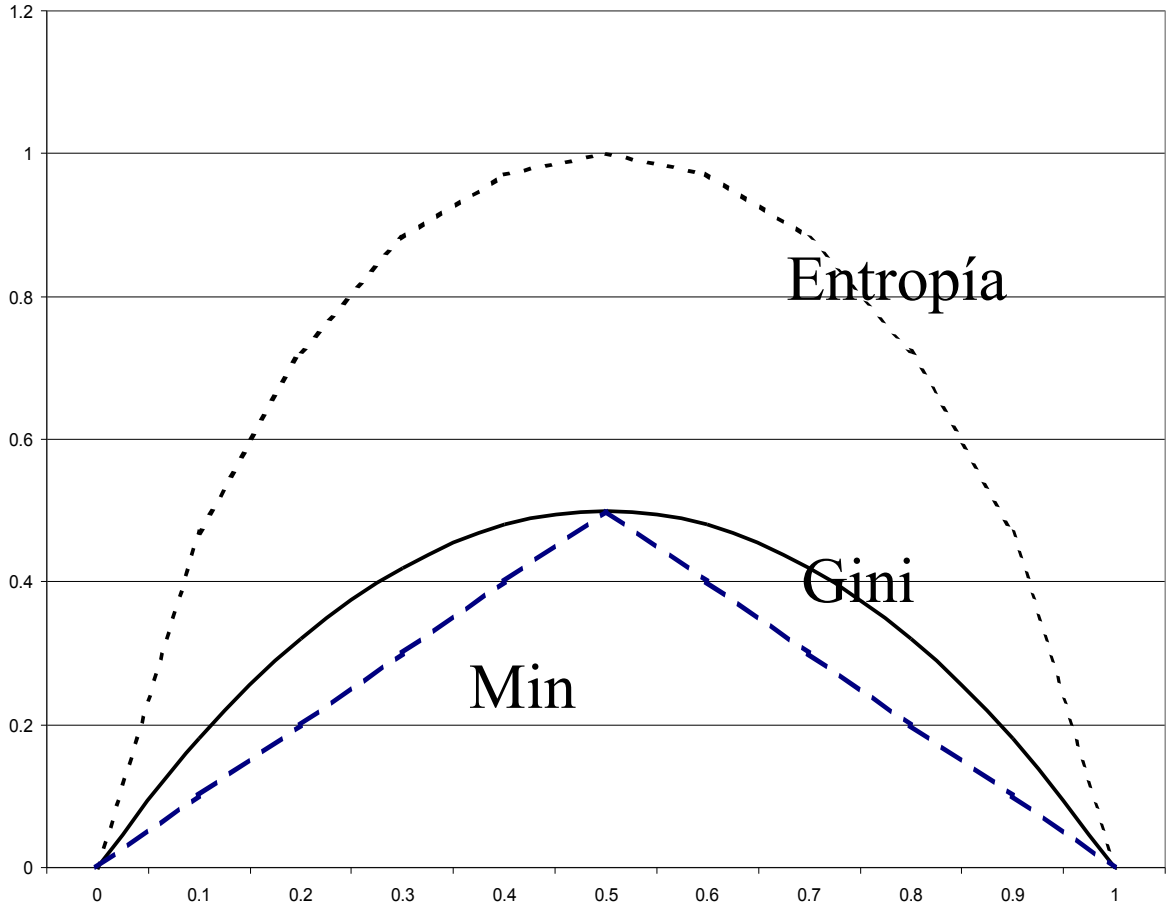
Una función $\phi : \Delta_k \rightarrow \mathbb{R}^+$ es una *medida de impureza* si satisface:

1. ϕ es mínima en los extremos de Δ_k
2. ϕ es máxima en el centro de Δ_k
3. ϕ no cambia si modificamos el orden de las componentes de un vector
4. ϕ es derivable en el interior de Δ_k

Indice Gini $\phi(x) = \sum x_i(1 - x_i)$

Entropía $\phi(x) = -\sum_i x_i \log x_i$

Entropía, Gini, Min

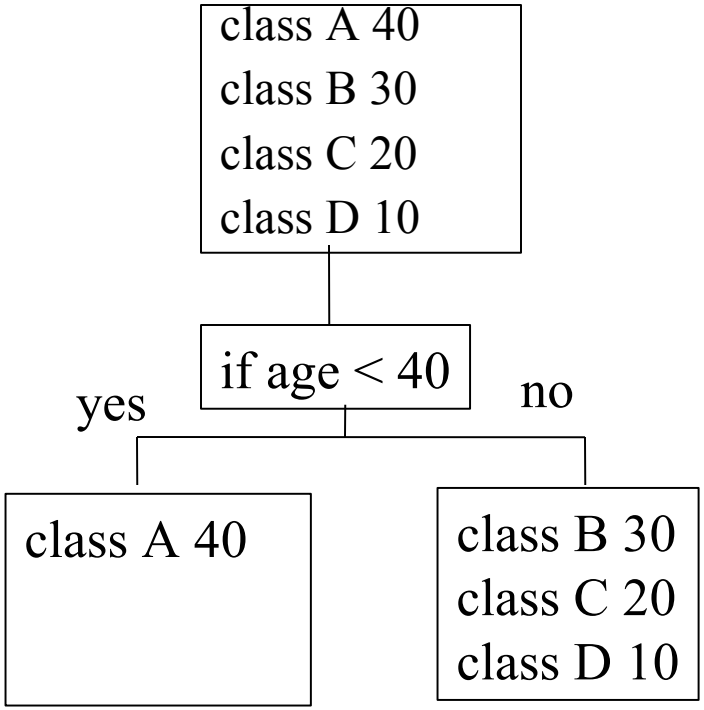


Entropía vs. Indice Gini

- Breiman, L., (1996).Technical note: Some properties of splitting criteria, *Machine Learning* 24, 41-47. Conclusiones de estudio empírico:
  - Gini tiende a seleccionar splits que ponen una clase mayoritaria en un sólo nodo y el resto en otros nodos (splits desbalanceados).
  - Entropía favorece splits balanceados en número de datos.

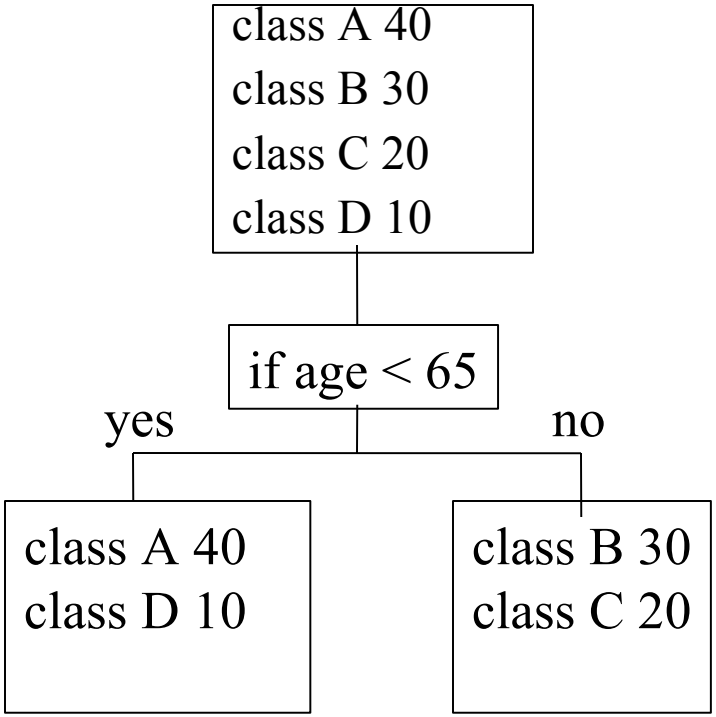
Entropía vs. Indice Gini

- Indice Gini tiende a aislar clases numerosas de otras clases



GiniSplit = 0,264 EntPond=0,259

- Entropía tiende a encontrar grupos de clases que suman más del 50% de los datos

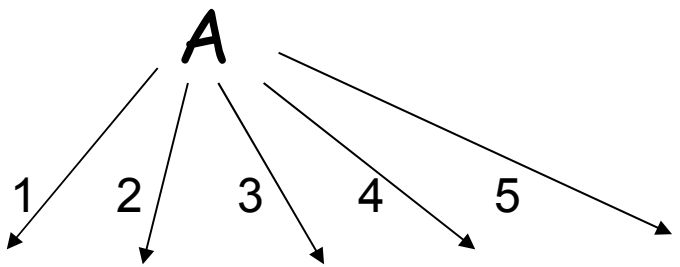


GiniSplit=0,4 EntPond=0,254

Problemas de entropía e Indice Gini

- Favorecen splits con muchos nodos hijos. Favorece atributos con muchos valores.

A	Clase
1	Si
2	No
3	No
4	Si
5	si



Problemas de entropía e Indice Gini

- Favorecen splits con muchos nodos hijos. Favorece atributos con muchos valores.
- Solución 1.: usar splits binarios
- Solución 2: usar GainRatio

$$GainRatio(t,S) = Gain(t,S) - SplitInfo(t,S)$$

$$SplitInfo(t,S) = Entropy(\frac{|s_1|}{|t|}, \frac{|s_2|}{|t|}, ..., \frac{|s_n|}{|t|})$$

Contenido

- Repaso algoritmo de Hunt
- Selección de Splits usando índice Gini
- Selección de Splits usando entropía
- Medidas de Impureza
- Selección de Splits usando test Chi-cuadrado
- Medidas de Separación

Selección de split usando test Chi-cuadrado

- El test de chi-cuadrado se usa para testear la hipótesis nula de que dos variables son independientes.
- Es decir la ocurrencia de un valor de una variable no induce la ocurrencia del valor de otra variable.

Notación básica de probabilidades

Los datos de entrenamiento son eventos instancias de las variables aleatorias $(\vec{X}, C)$. $\vec{X} = (X_1, \ldots, X_n)$

Denotamos $P(X_1 = x_1, \ldots, X_n = x_n, C = c)$ al número de eventos donde $X_1 = x_1, \ldots, X_n = x_n, C = c$ dividido por el número de eventos totales.

Nota: esto es una suposición ya que en general
los datos de entrenamiento son una muestra de
la población total sobre la que se definen las probabilidades.

Notación básica de probabilidades (cont.)

En general, para dos variables W, Y, la *probabilidad condicional* $P(W = w \mid Y = y)$ se define como

$$\frac{P(W=w, Y=y)}{P(Y=y)}.$$

Dos variables W, Y son independientes si $P(W = w, Y = y) = P(W = w)P(Y = y)$, para todo x, y en el dominio de las variables. Es decir tenemos que $P(W = w \mid Y = y) = P(W = w)$ y lo mismo para Y.

Ejercicio: probabilidades para weather.nominal

- El vector de variables es (Outlook,Temp,Humidity,Windy,Play)
- Asumiendo que los datos corresponden a las probabilidades reales (esto no es siempre así ¿por qué?)
- La variable de la clase es Play.
- Calcule las siguientes probabilidades:
  - P(Play=yes) P(Play=no)
  - P(Play = yes, Outlook= sunny), P(play=no, Outlook=overcast)
  - P(Outlook,Temp,Humidity) para todas las combinaciones.
  - P(Play=yes | Windy=false)
- Haga el cálculo para establecer si Outlook y Play son independientes
- Haga el cálculo para establecer si Temp y Play son independientes

Test de Chi-cuadrado

- Basado en Noción de tabla de contingencia
  - Nos sirve para analizar si dos variables son dependientes
- Ejemplo: venta de té y café en un supermercado.
  - Tenemos dos variables Café (C) y Té (T)
  - Valores son compró o no-compró

Tabla de Contingencia

	c	$\overline{c}$	Total
t	20	5	25
$\overline{t}$	70	5	75
Total	90	10	100

Tabla de Contingencia (cont.)

- Cada celda $r=(r_1,\ldots,r_n)$, contiene el número de veces $O(r)$ que ocurren juntos en los datos los valores $r_1,\ldots,r_n$
- Para una celda $r=(r_1,\ldots,r_n)$, definimos el valor esperado de la celda si las variables son dependientes E(r):

$$E[r] = n \times \frac{E[r_1]}{n} \times \ldots \times \frac{E[r_k]}{n}$$

donde $E(r_i)=O(r_i)$ es el número de ocurrencias de r_i en la tabla

Ejemplo: valor esperado E

	c	$\overline{c}$	Total
t	20	5	25
$\overline{t}$	70	5	75
Total	90	10	100

$$E[t\overline{c}] = 100 \times \frac{25}{100} \times \frac{10}{100} = 2.5$$

$$E[tc] = 100 \times \frac{25}{100} \times \frac{90}{100} = 22.5$$

Estimador Chi-cuadrado

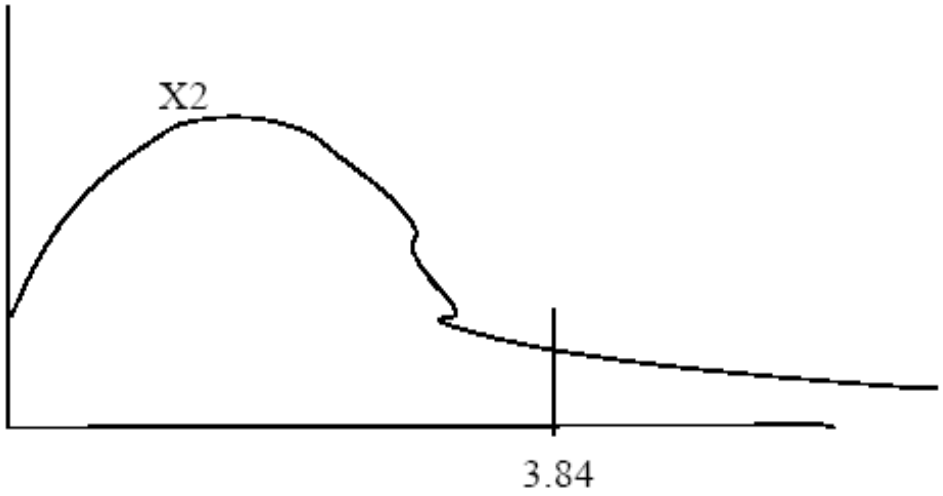
Estimador Chi-Cuadrado:

$$\mathcal{X}^2 = \Sigma_{r \in R} \frac{(O(r) - E[r])^2}{E[r]}$$

Desviación normalizada entre la tabla de contingencia real y la esperada

Estimador Chi-cuadrado (cont.)

Si $\mathcal{X}^2$ es mayor que 3.84 rechazamos la suposición de independencia con un 95% de nivel de confianza.



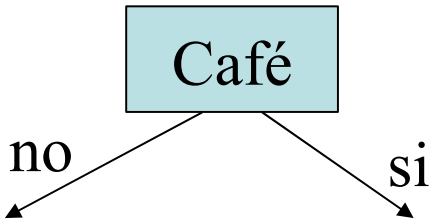
Grados de libertad = (#cols−1) (#filas−1)

Uso de Chi-cuadrado para seleccionar splits

- Consideremo un split binario sobre una variable *A*.
- Las variables para la tabla de contingencia son *C* (variable de la clase) y *A*
  - Cada nodo hijo representa un valor distinto para *A*
- Se calcula tabla de contingencia y estimador chi-cuadrado
- Se elige el split con el mejor estimador chi-cuadrado

Ejemplo

- Supongamos que el atributo *A* es Café (*C*), y la variable de la clase es Té (T)
- El split quedaría



Ejercicio: selección de splits usando chi-cuadrado en weather.nominal

1. Calcule el estimador chi-cuadrado para los splits simples de Windy y Humidity
2. Diga en base al test chi-cuadrado si estos splits son independientes de la variable de la clase
3. Calcule el estimador chi-cuadrado para el split simple sobre Outlook.

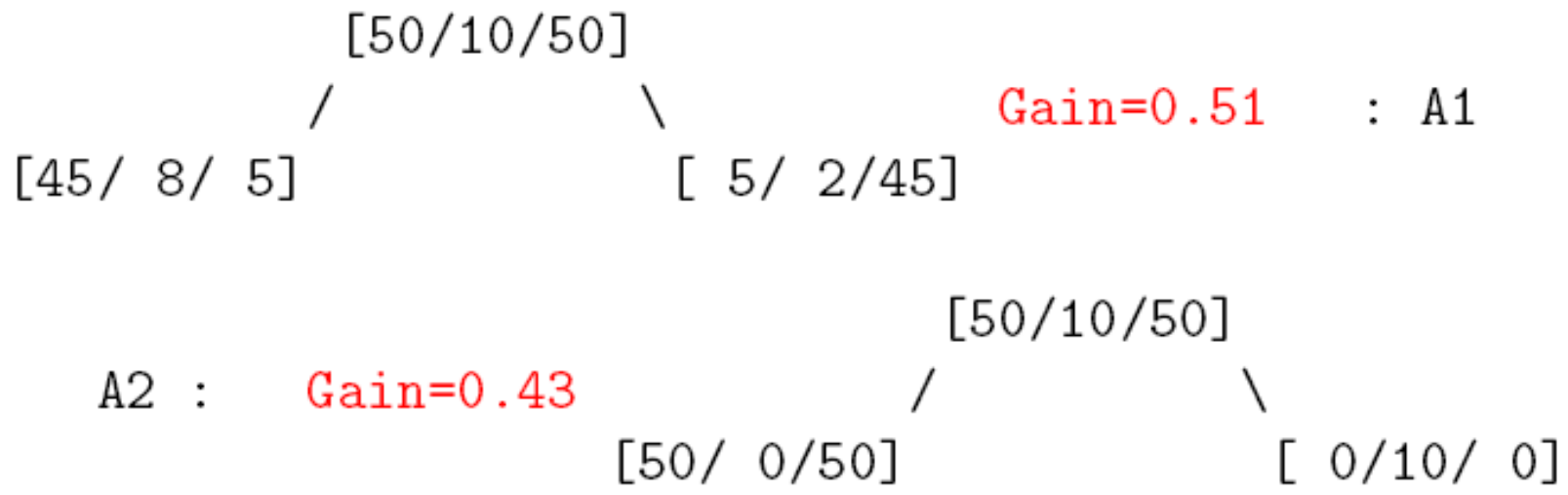
Efecto de distintas medidas de impureza

- A pesar de lo reportado por Breiman, un estudio de Mingers (1989) cocluye:
  - Los árboles obtenidos con distintas criterios de selección de splits no difieren significativamente en su poder predictivo
  - El criterio sí afecta el tamaño del árbol obtenido
  - La elección aleatoria de atributos produce árboles de alrededor del doble de tamaño que si utilizamos una medida de impureza
  - Entre las medidas vistas, Gain ratio produce árboles pequeños y Chi-cuadrado produce árboles grandes

Problema de las medidas de impureza

- Fayyad e Irani (1993): no son capaces de preferir splits que separan clases

sean A_1, A_2 dos atributos tales que inducen los siguientes splits de este nodo:



Contenido

- Repaso algoritmo de Hunt
- Selección de Splits usando índice Gini
- Selección de Splits usando entropía
- Medidas de Impureza
- Selección de Splits usando test Chi-cuadrado
- Medidas de Separación

Medidas de Separación

Vemos que las medidas de impureza no son capaces de preferir splits que separan clases. Por esto, se propone en Fayyad e Irani('93) una nueva familia de medidas, las medidas de separación (C-SEP). Sea S un conjunto de ejemplos S, y S_1, S_2 un split de S (split binario).

Medidas de Separación

Una medida de separación cumple las siguientes propiedades:

1. Su valor es máximo cuando el set de clases de elementos en S_1 es disjunto del de S_2
2. Es mínimo cuando la distribución de clases de S_1 y S_2 son iguales
3. Favorece splits que mantienen ejemplos de la misma clase en el mismo nodo
4. Es sensible a los cambios en las distribuciones de clases
5. Es positiva, derivable e insensible a las permutaciones de clases

Medida de Ortogonalidad

$$\mathrm{ORT}(S_1, S_2) = 1 - \cos\theta(V_1, V_2)$$

Esta medida de ortogonalidad entre S_1 y S_2 sí prefiere el split intuitivamente apropiado en el ejemplo anterior. Además, se comporta empíricamente (tests de prueba reales) mejor que algoritmos basados en entropía, basado en criterios de tasas de error y número de hojas.

Referencias

(no son obligatorias para el curso)

L. Breiman.
Technical note: some properties of splitting criteria.
Machine Learning, 24:41–47, 1996.

J. Mingers.
An empirical comparison of selection measures for decision-tree induction.
Machine Learning, 3:319–342, 1989.

S. Rogic.
An overview of methods for decision tree induction.
Term project for CS 532B.

K.V. Sreerama.
On growing better decision trees from data.
PhD thesis, The Johns Hopkins University, 1995.