Arboles de Decisión (I)

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

Clasificación

- Dado un conjunto de datos:

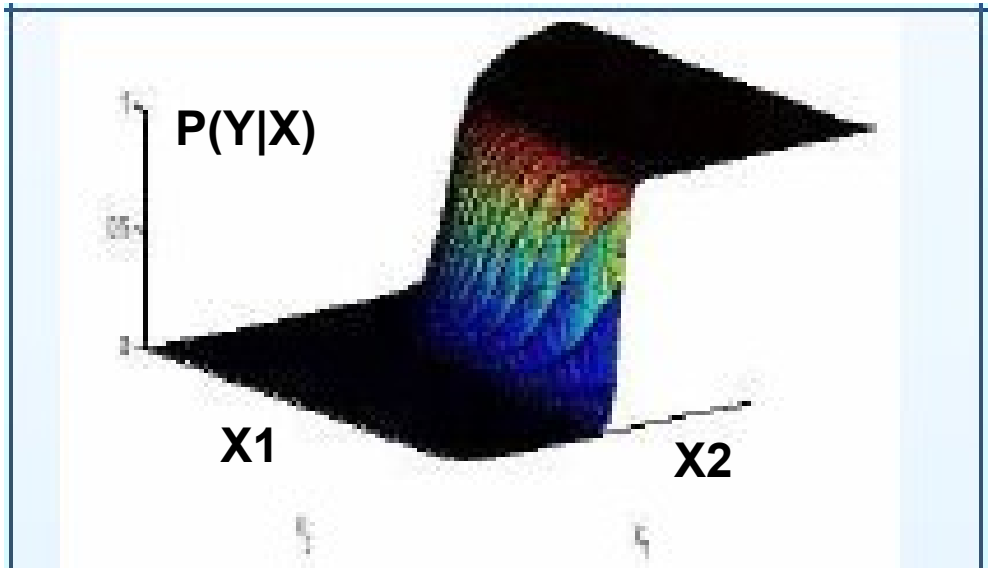
$$D = \{(x1,y1),(x2,y2),\ldots, (xn,yn)\}$$

que son instancias de un vector X y variable categórica Y (variable de la clase).

- Construir un modelo (función) f(X) para predecir Y.

Sistema observado

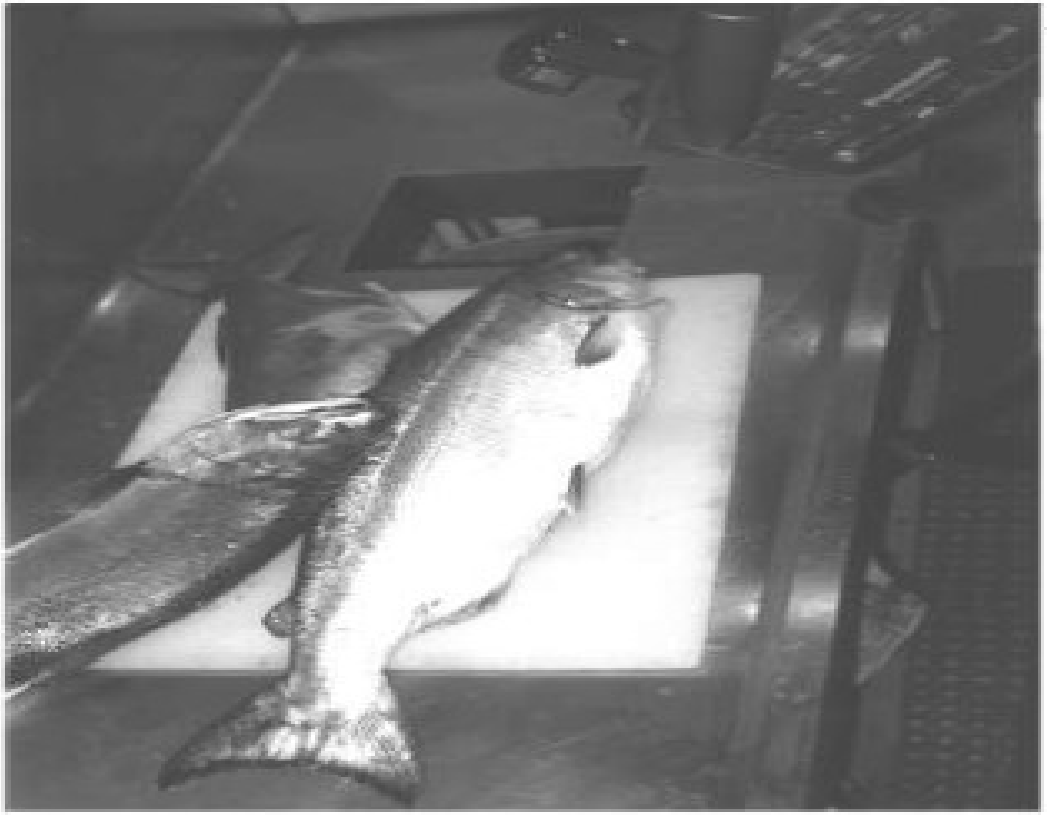
- Los datos son observaciones de un sistema de variables X e Y
- La variable Y no es necesariamente función de X
- X,Y ~ P(X,Y)
- X,Y ~ P(X), P(Y|X)

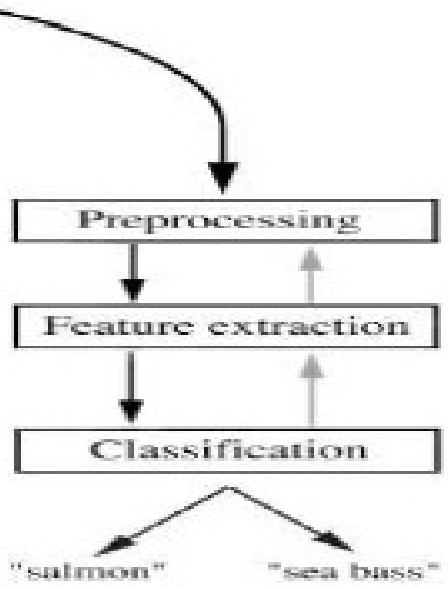


Etapas en Clasificación

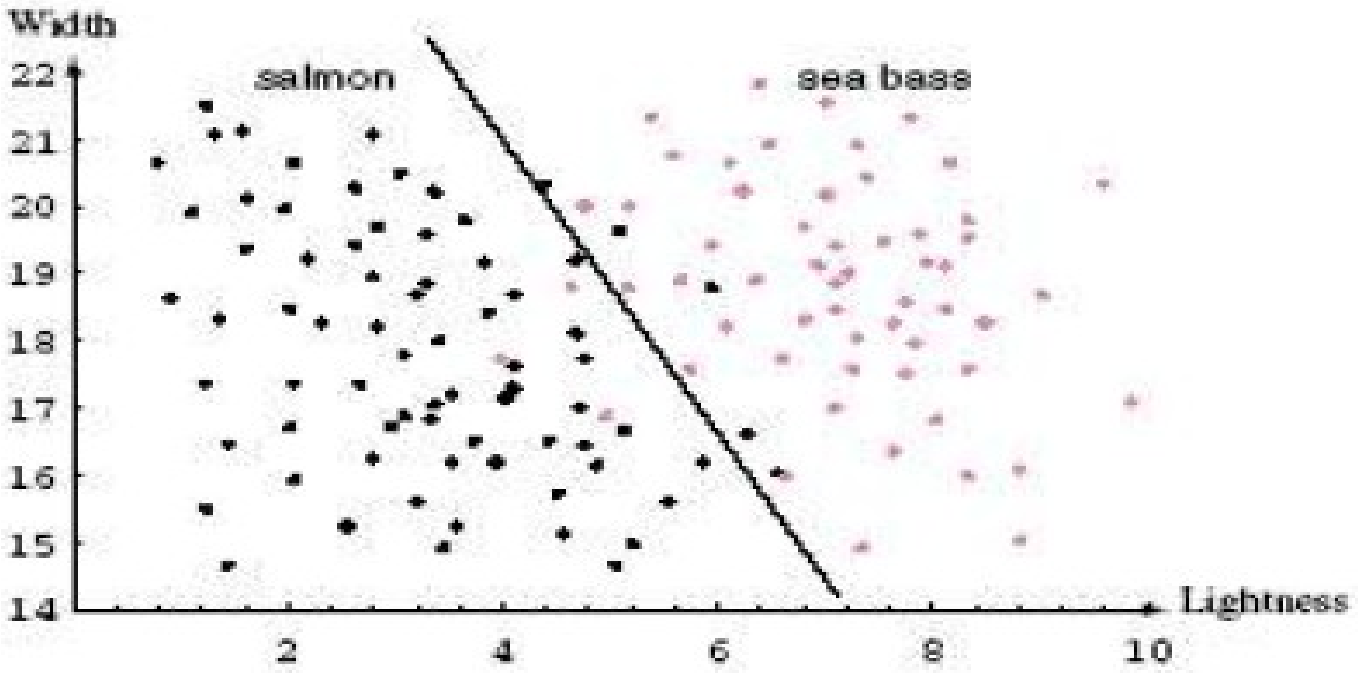
- Construcción del modelo
  - Datos de entrenamiento
- Prueba del modelo
  - Datos de prueba
- Uso del modelo
  - Datos de uso

Clasificación: ejemplo





Clasificación: ejemplo (cont.)



- $f(X) = \begin{cases} 1 & \text{si } W + 4{,}5\, L - 39 < 0 \\ 0 & \text{si no.} \end{cases}$

Aplicaciones de Clasificación (I)

- Detección de Fraude:
  - Objetivo: predecir uso fraudulento en tarjetas de crédito
  - Método:
    - Usar transacciones de compras en un determinado período e info. en cuentas.
    - Definir atributos de entrada como: cuándo se compra, frecuencia de compra, frecuencia de pagos, etc.
    - Inducir un modelo para predecir clase de nuevas transacciones de compra.

Aplicaciones de Clasificación (II)

- Predicción de Deserciones ("churn"):
  - Objetivo: predecir qué clientes abandonarán un determinado servicio (ej., teléfono celular)
  - Método:
    - Definir atributos de entrada a partir de transacciones del cliente (ej., llamados telefónicos, frecuencia, último llamado, interrupciones, etc.) e info. descriptiva de usuarios.
    - Usar info. de meses anteriores y definir atributo de clase como abandono o no abandono del servicio.
    - Inducir un clasificador.

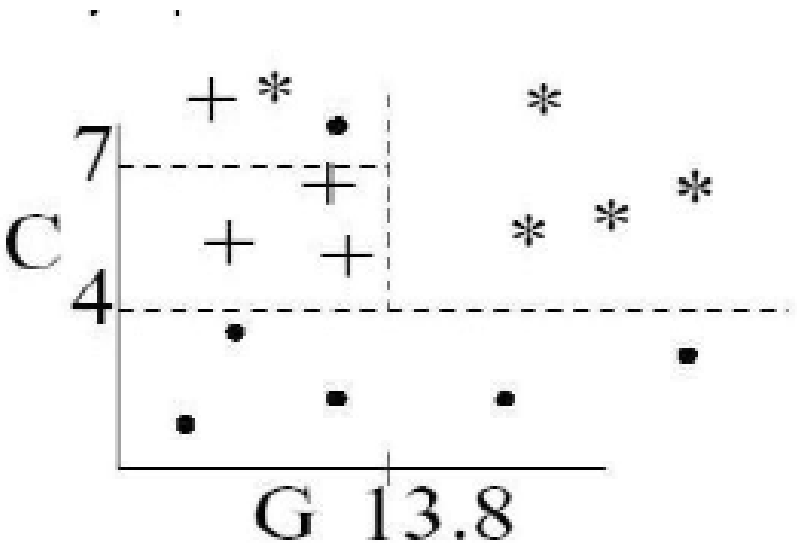
Aplicaciones de Clasificación (III)

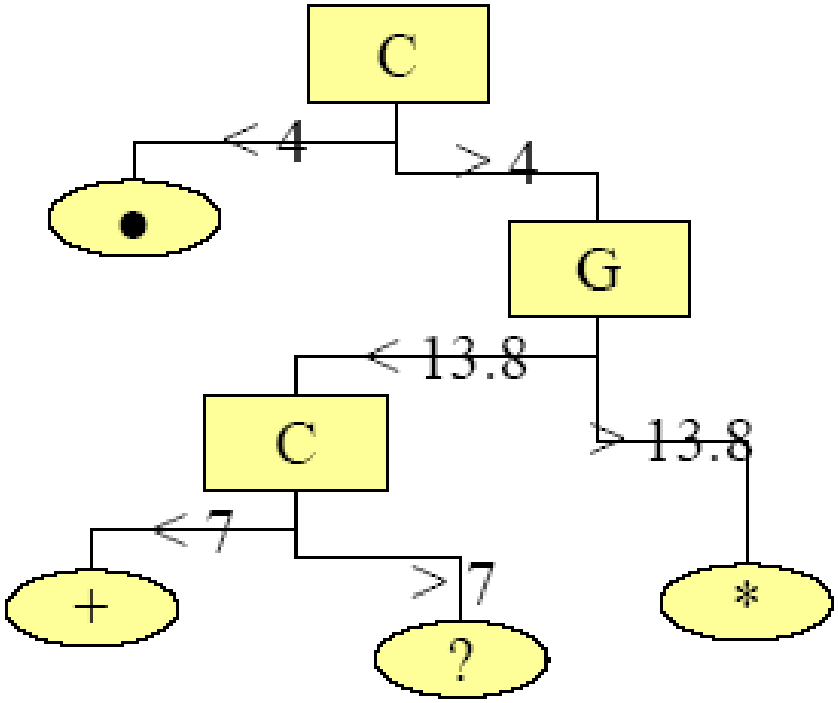
- Clasificación de Documentos:
  - Objetivo: Predecir tópico de un nuevo documento
  - Método:
    - Definir atributos de entrada a partir del contenido del documento (e.g., vector de términos)
    - Usar documentos pre-clasificados como datos de entrenamiento.
    - Inducir un clasificador.

Clasificación: Tipos de Modelos

- Enfoque Discriminante
  - Arboles de Decisión
  - Reglas de Decisión
  - Discriminantes lineales
- Enfoque Generativo
  - Redes Bayesianas
  - Modelos paramétricos
- Enfoque de Regresión
  - Redes Neuronales
  - Regresión Logística

Arboles de Decisión





Arboles de Decisión

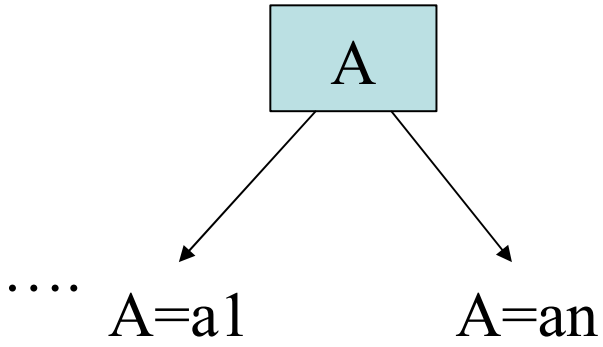
- Fáciles de construir
- Fáciles de interpretar
- Buena precisión en muchos casos

Construcción de árboles de decisión: algoritmos

- Algritmos de Memoria Principal
  - Algoritmo de Hunt (CLS, 1960´s)
  - ID3 (Quinlin 70's and 80's), C4.5 (Quinlin 90's)
- Algritms Escalables
  - SLIQ, SPRINT

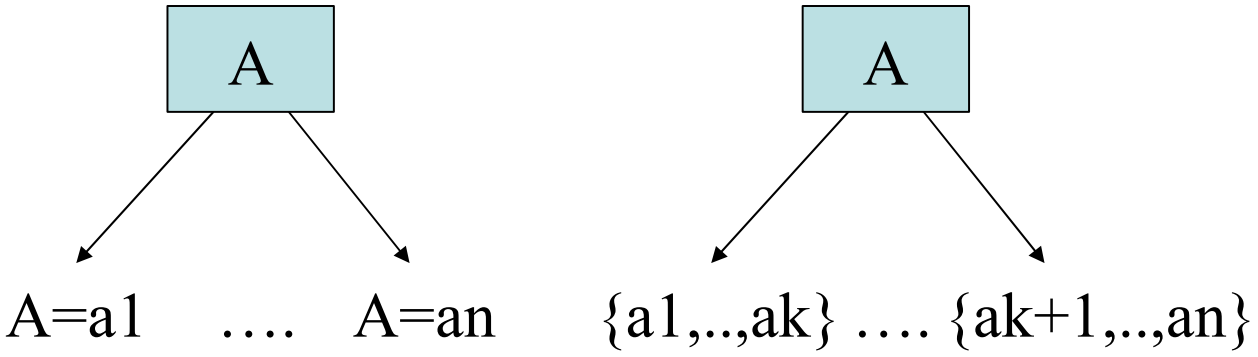
Split

- Definición: un *split* es una variable (atributo) más una lista de condiciones sobre la variable.



Split para variables categóricas

- Split Simple vs. Split Complejo



Split para variables numéricas

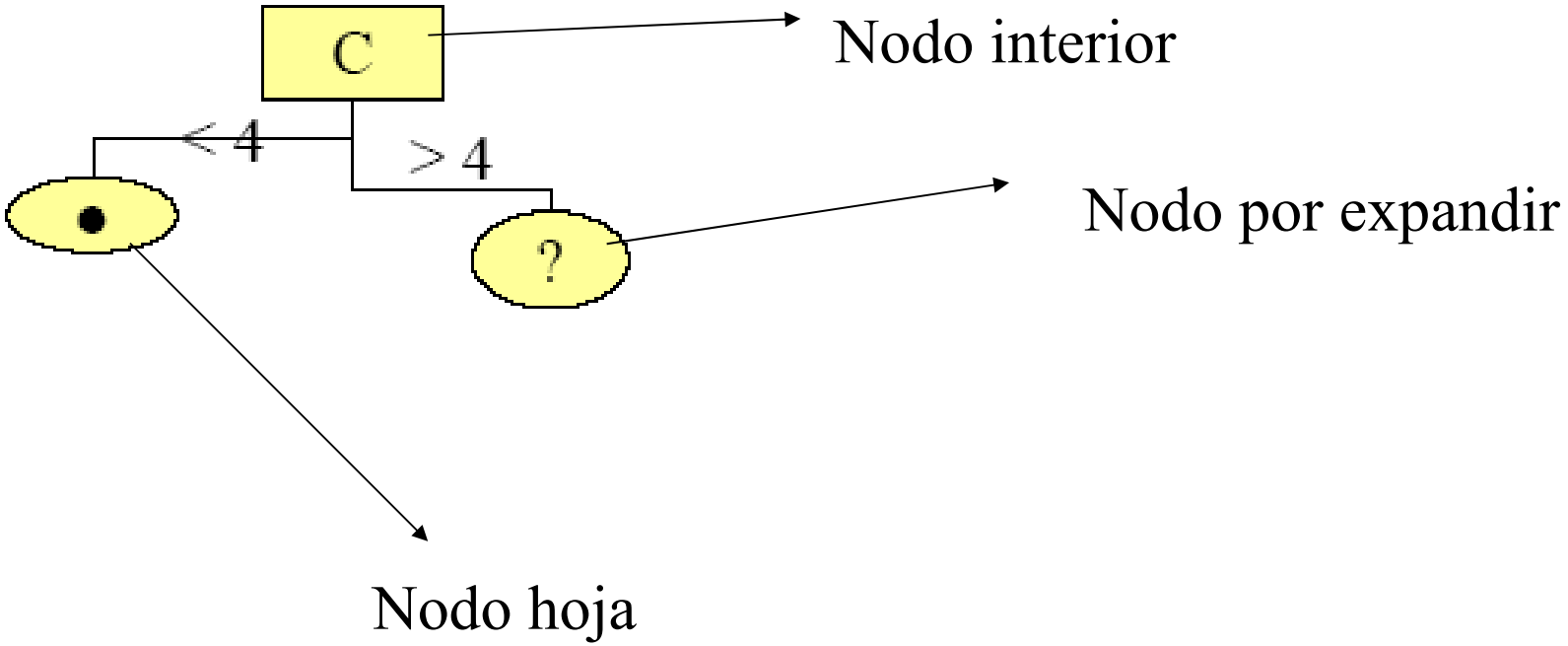
- Decisión binaria: $A <= v$, $A > v$
- Rangos: [0,15k), [15k,60k), [60k,100k), etc.
- Combinación lineal de variables

$$3A + 4B > 5$$

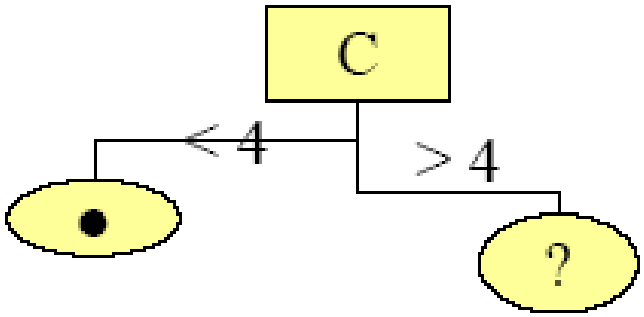
Algoritmo de Hunt

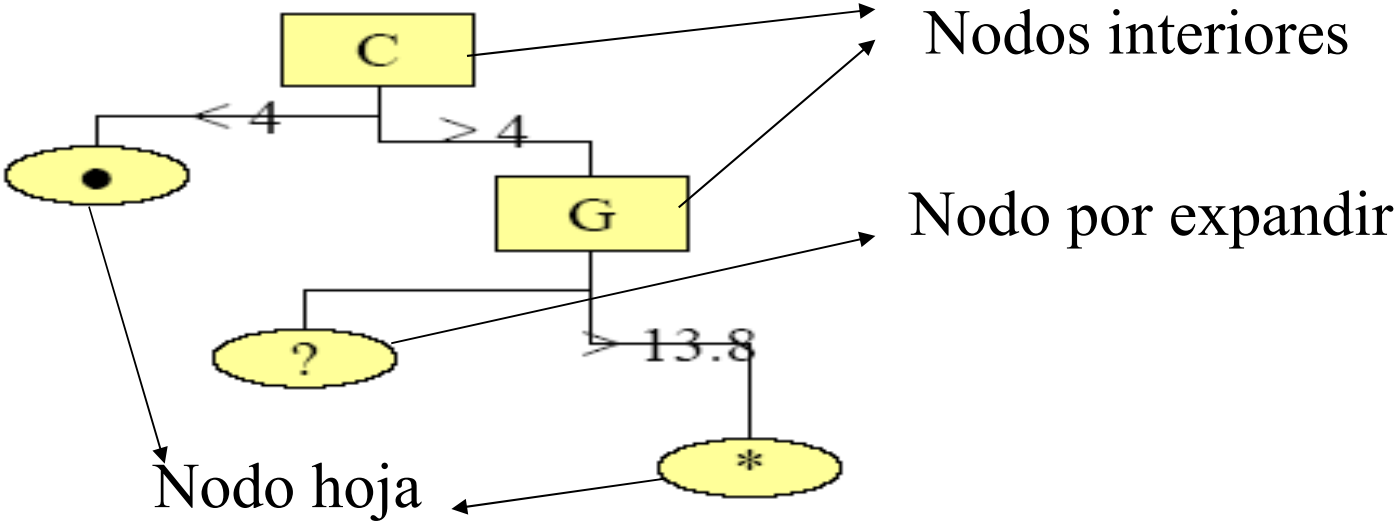
- Idea básica: cada nodo en el árbol de decisión tiene asociado un subconjunto de los datos de entrenamiento
- Inicialmente, el nodo raíz tiene asociado todo el conjunto de entrenamiento
- Construimos un árbol parcial que tiene tres tipos de nodos:
  - Expandidos (interiores)
  - Hojas: serán hojas en 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)

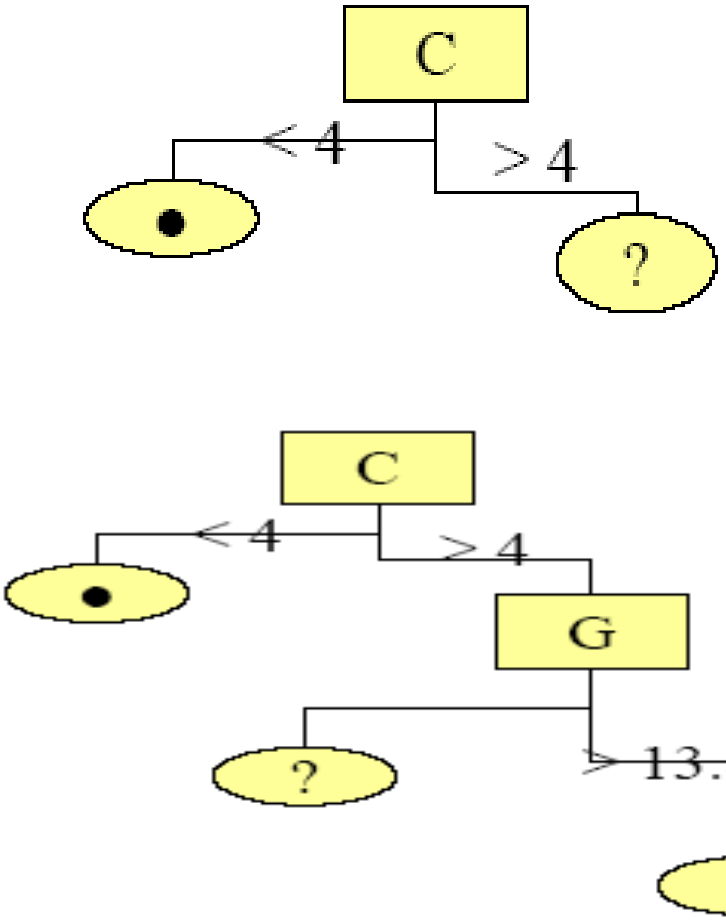


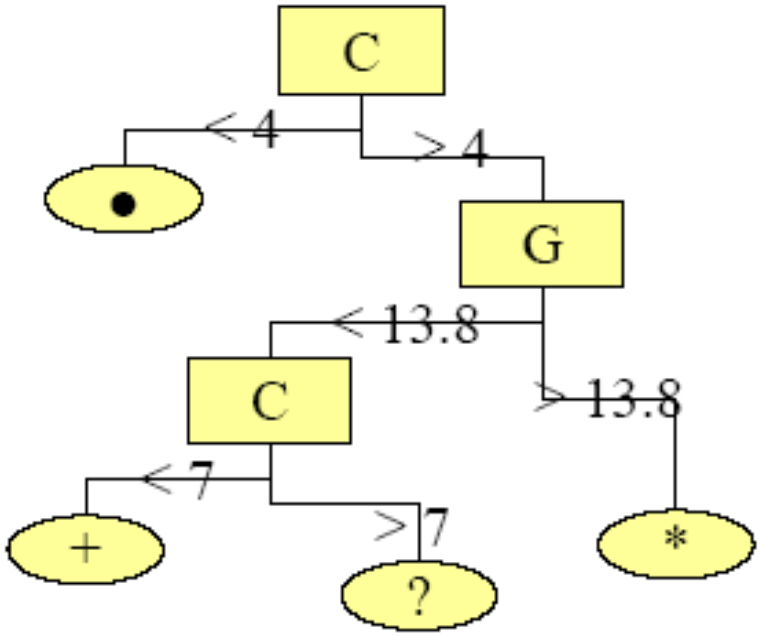
Algoritmo de Hunt (II)





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

- Las operaciones de expansión se realizan "primero en profundidad"
- Lo complejo es encontrar el mejor split en cada operación de expansión
- Número de splits a buscar depende del tipo de split que consideramos.

¿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

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

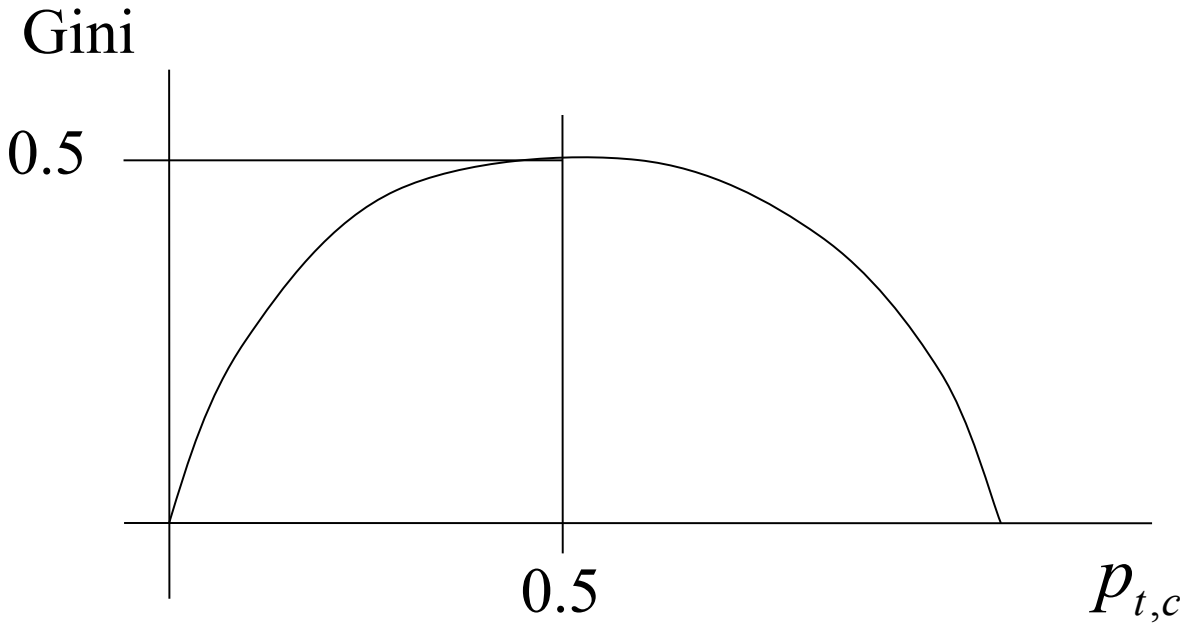
$$Gini(t) = 1 - \sum_{c \in C} p_{t,c}^{\ 2}$$

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

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

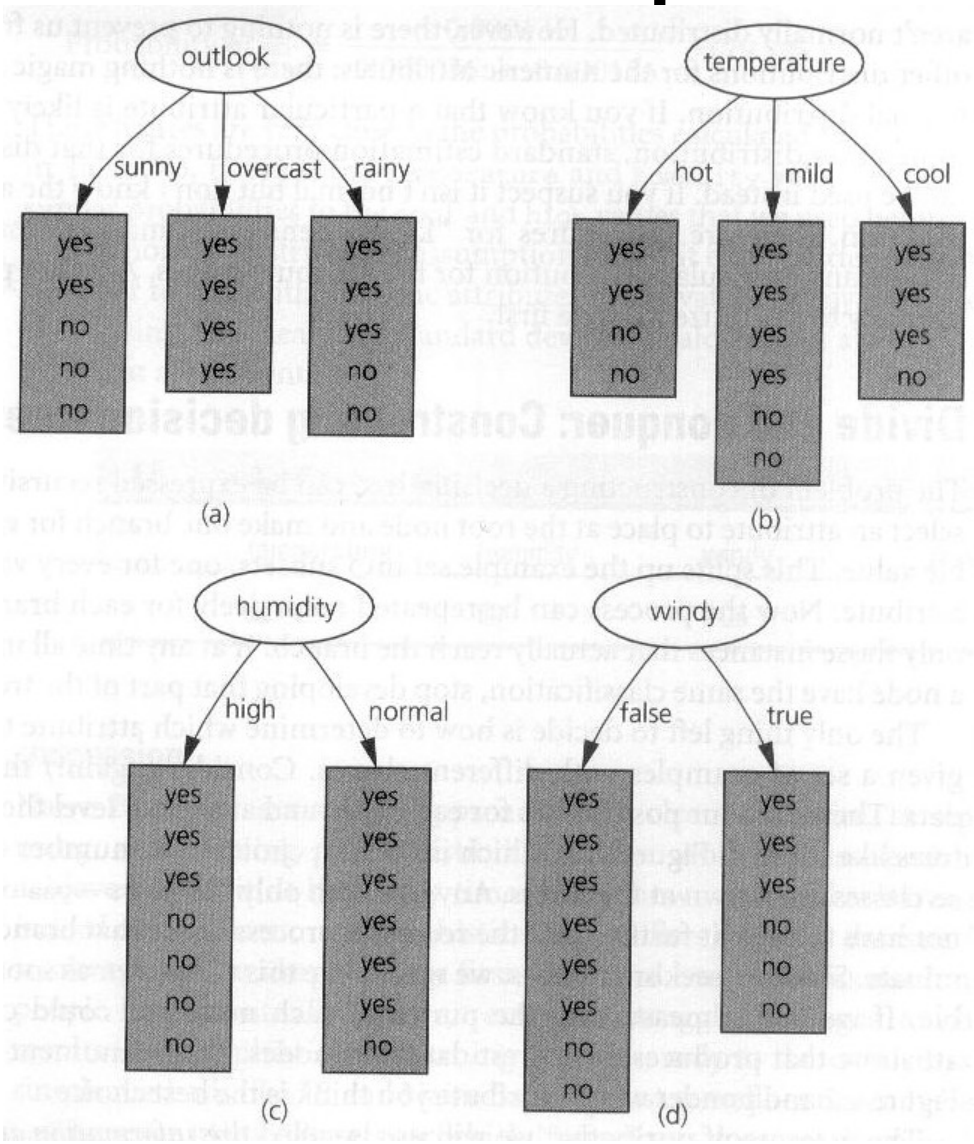
Ejemplo: weather.nominal

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

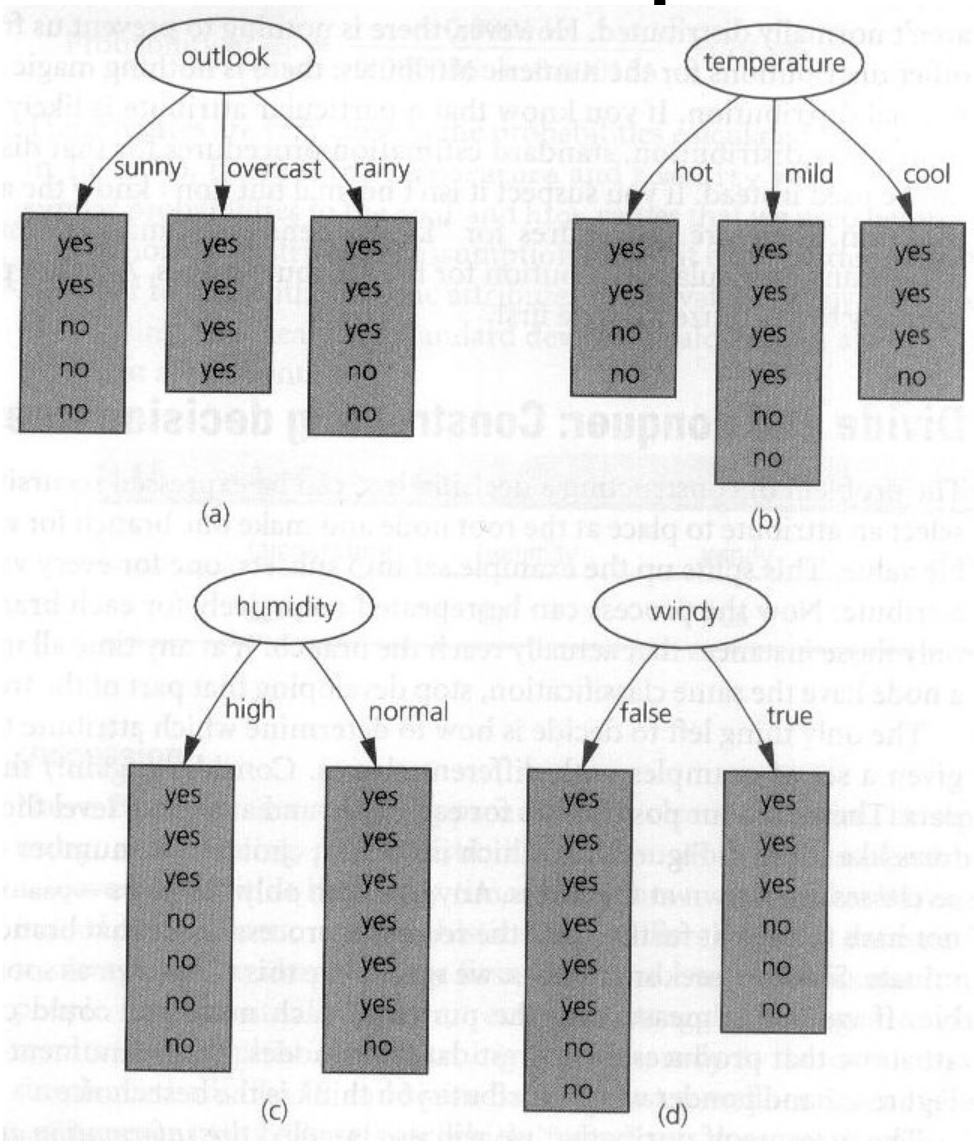
Posibles splits



weather.nominal: atributos vs. clases

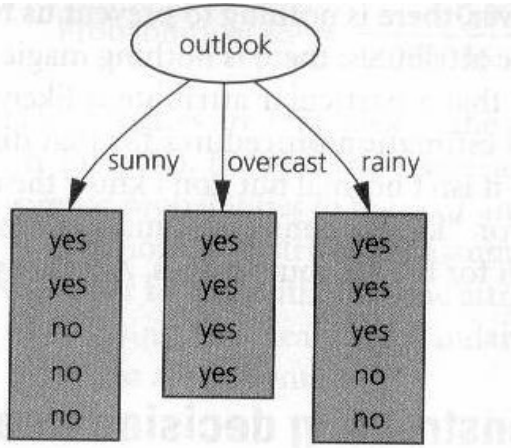
Outlook			*Temperature*			*Humidity*			*Windy*			*Play*	
	Yes	*No*		*Yes*	*No*		*Yes*	*No*		*Yes*	*No*	*Yes*	*No*
Sunny	2	3	Hot	2	2	High	3	4	FALSE	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	TRUE	3	3		
Rainy	3	2	Cool	3	1								

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

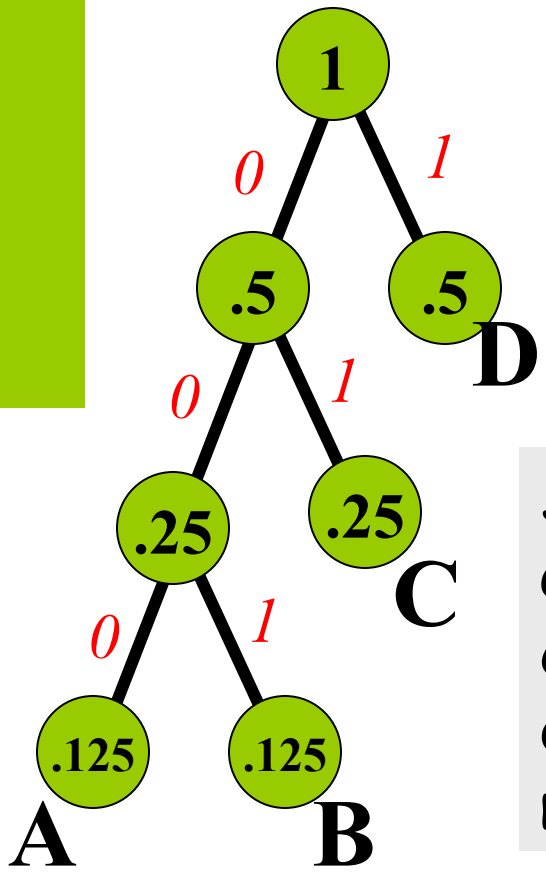
1. Dar el número de splits que necesitamos evaluar en la primera iteración del algoritmo de Hunt, para (a) splits complejos y (b) splits simples.
2. Seleccionar el mejor split usando el criterio del índice Gini
   - Calcular el GiniSplit de cada split
   - Determinar el mejor split

Teoría de Información

- Supongamos que transmitimos en un canal (o codificamos) mensajes que son símbolos en {x1,x2,…,xn}.
- A primera vista, necesitamos log n bits por mensaje.
- Si el emisor y el receptr cncen la distribucin de prb. P(X=xi)=pi, podríamos elabrar un cdg que requiera menos bits por mensaje.

Códigos de Huffman

Símb.	Prob.
A	.125
B	.125
C	.25
D	.5



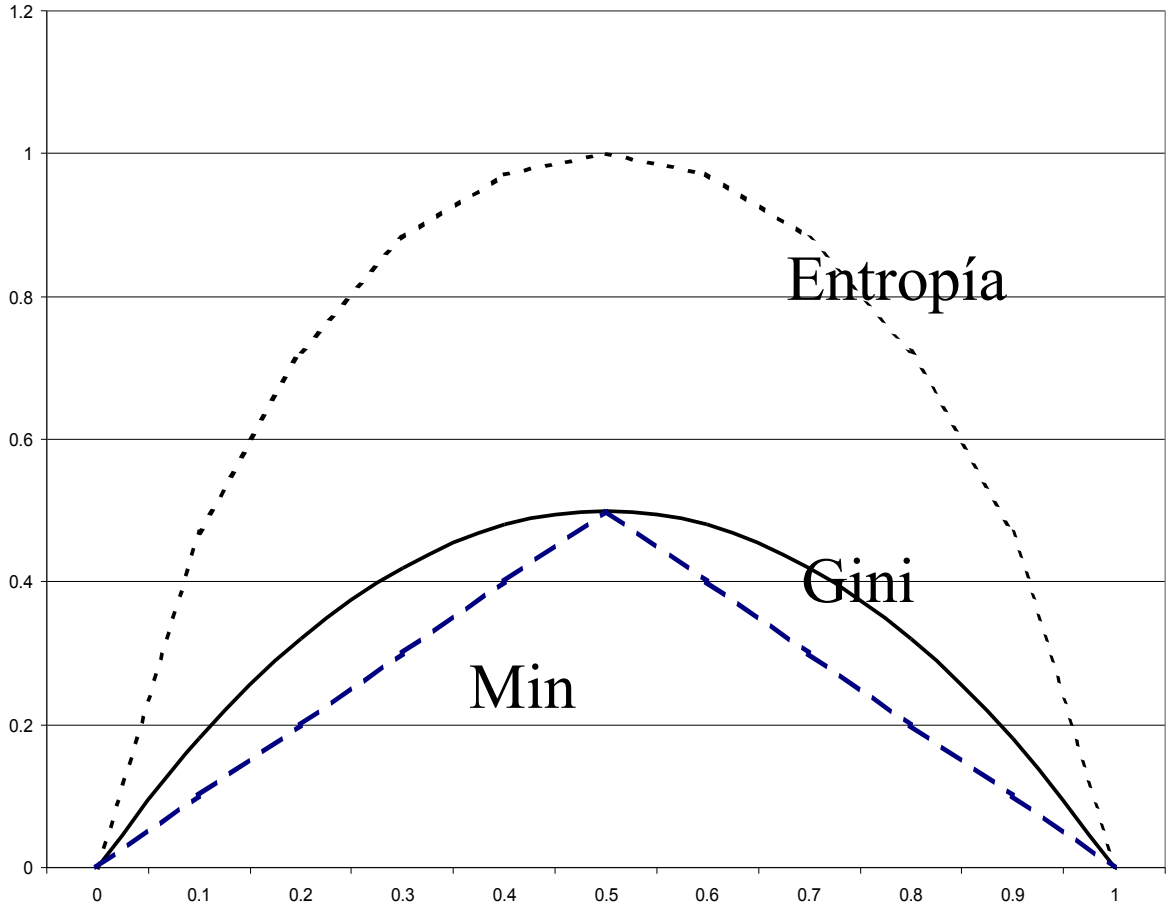
M	cod.	long.	prob	
A	000	3	0,125	0,375
B	001	3	0,125	0,375
C	01	2	0,250	0,500
D	1	1	0,500	0,500
average message length				1,750

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

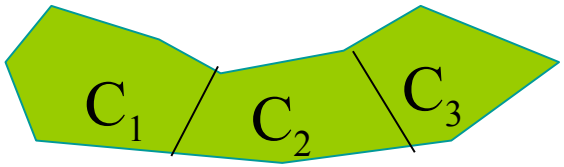
Entropía



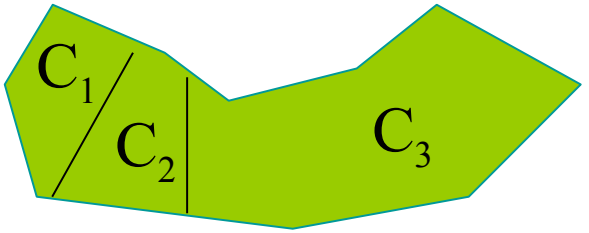
Selección de splits usando Entropía

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

- 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



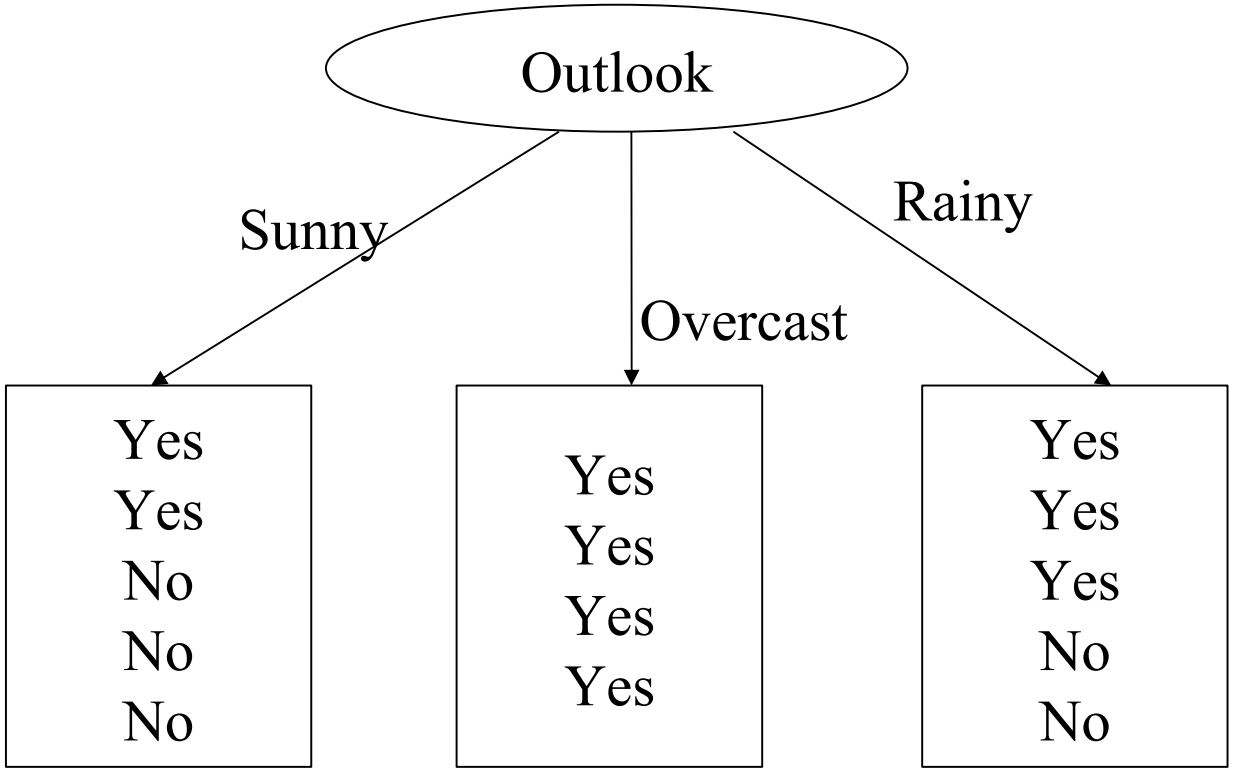
Entropía baja

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

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

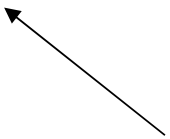
$$= 0.92 - \frac{5}{14} \cdot 0.97 - \frac{4}{14} \cdot 0 - \frac{5}{14} \cdot 0.97$$

$$= 0.92 - 0.69 = 0.23$$

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

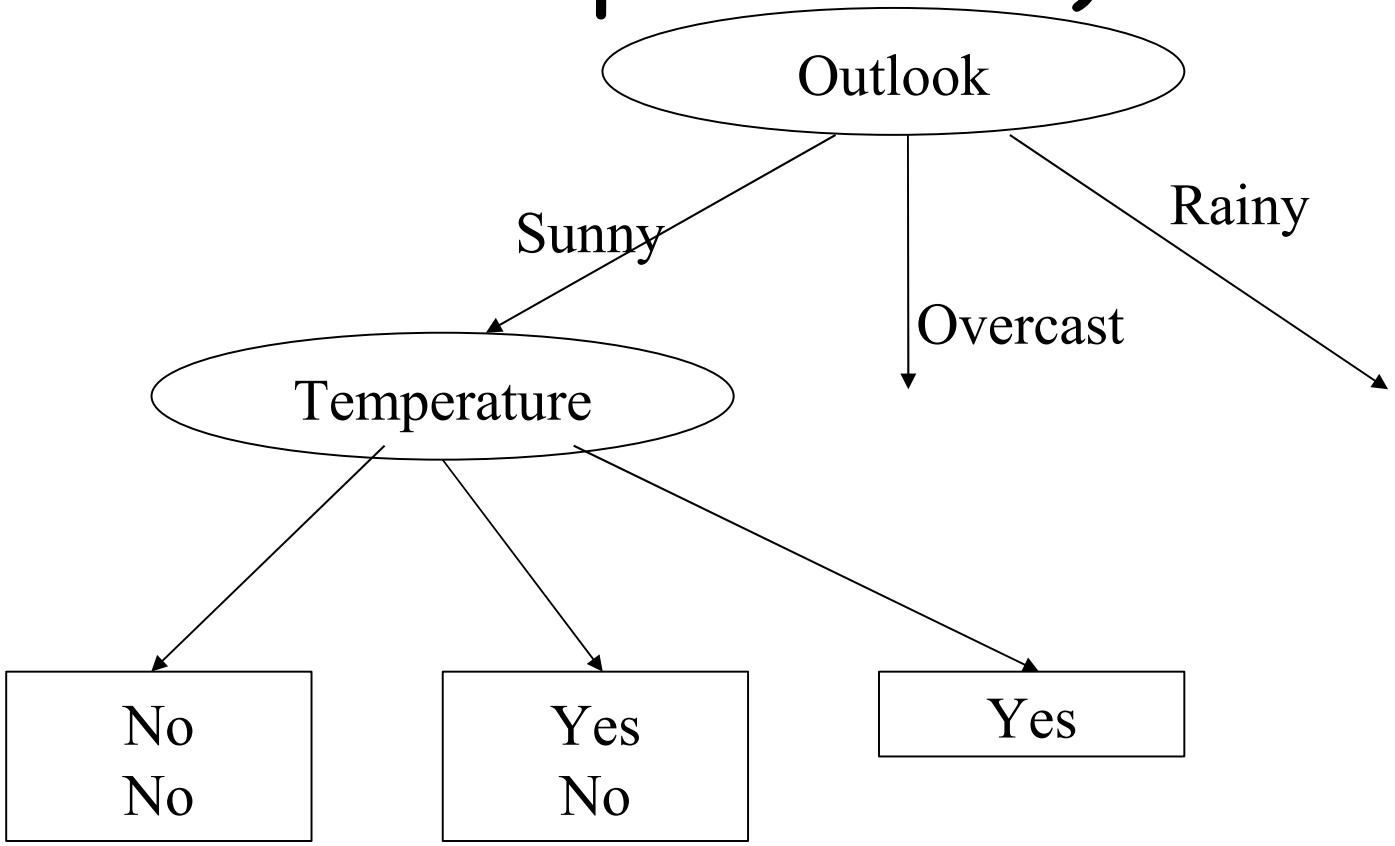
$$Gain(S, humidity) = 0.15$$

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



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"

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

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

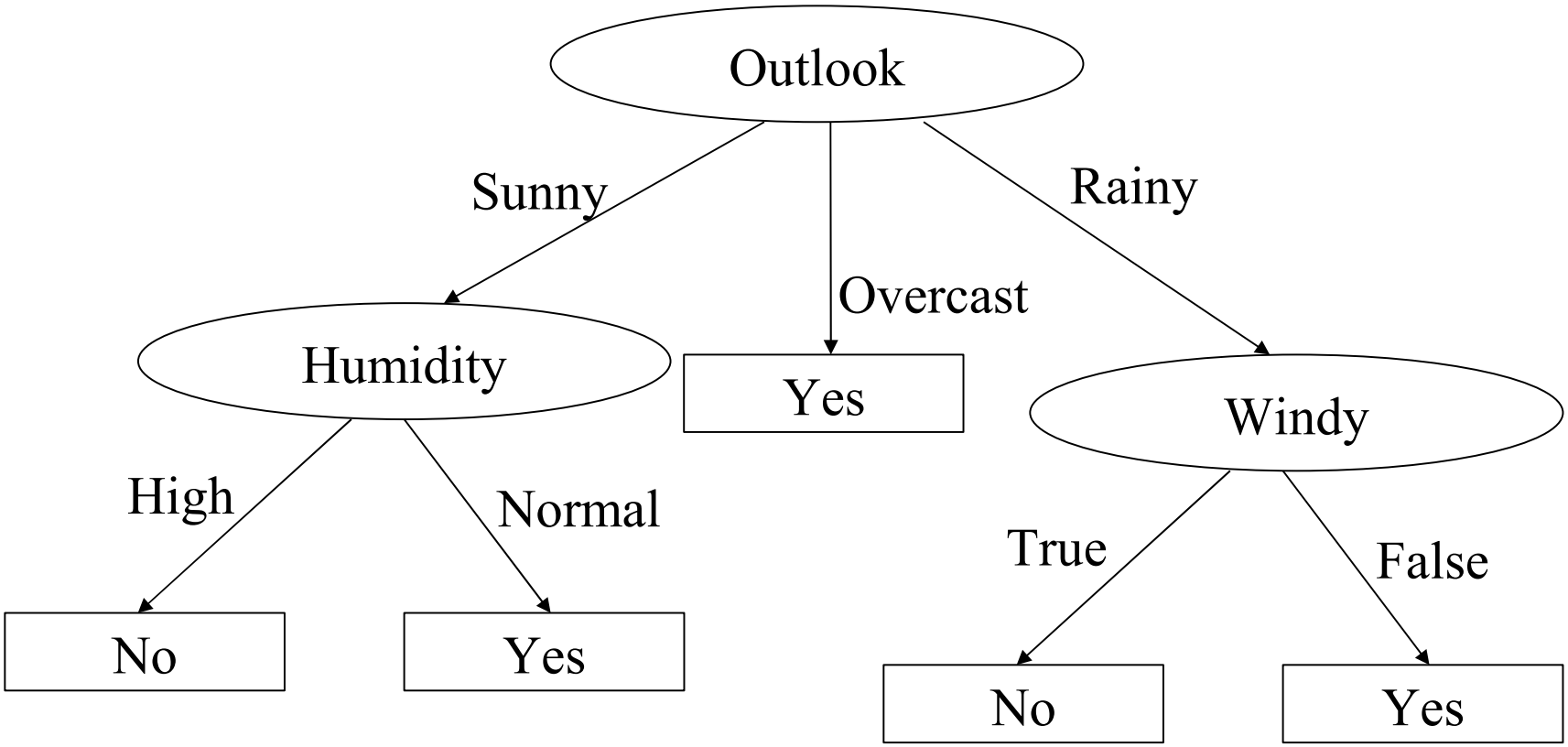
$$= 0.97 - 0.40 = 0.57$$

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

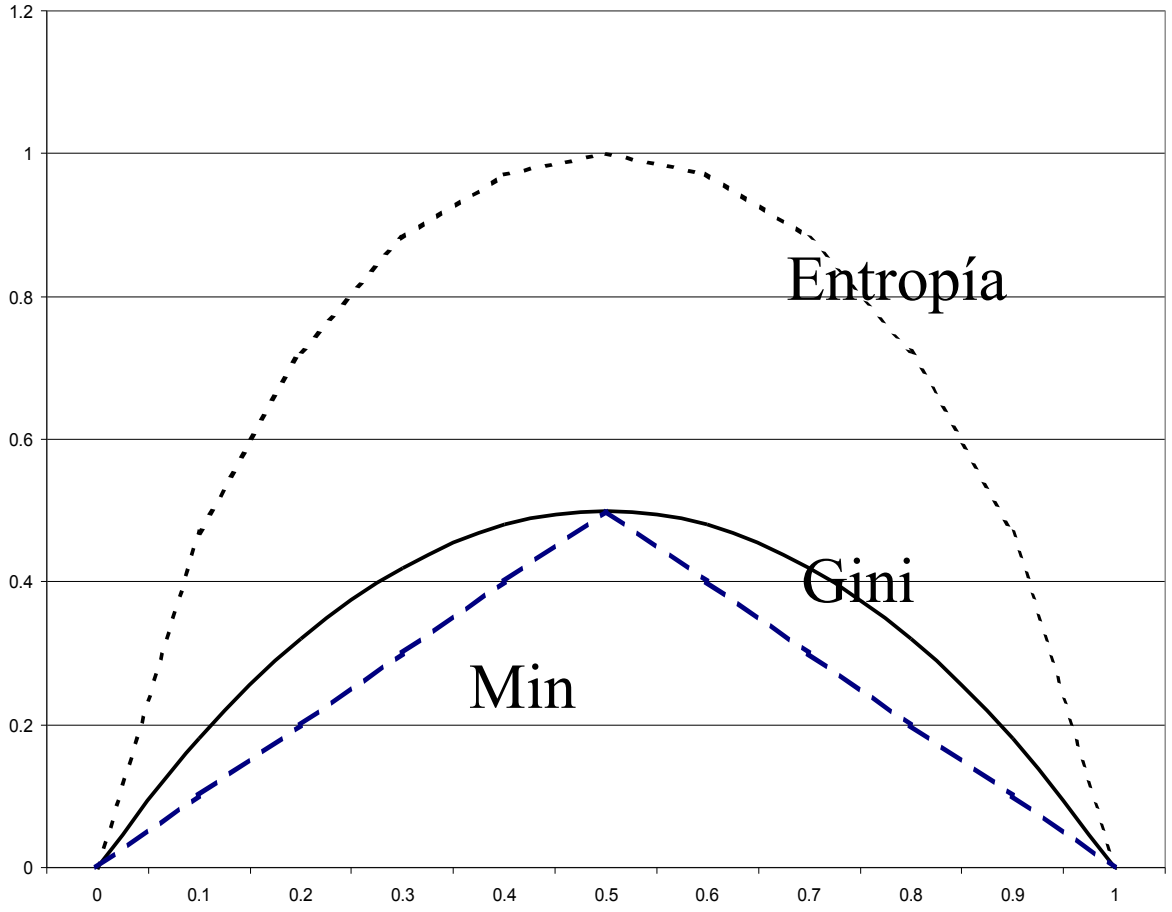
$$Gain(S, windy) = 0.02$$

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

Ejemplo: Arbol resultante



Entropía, Gini, Min

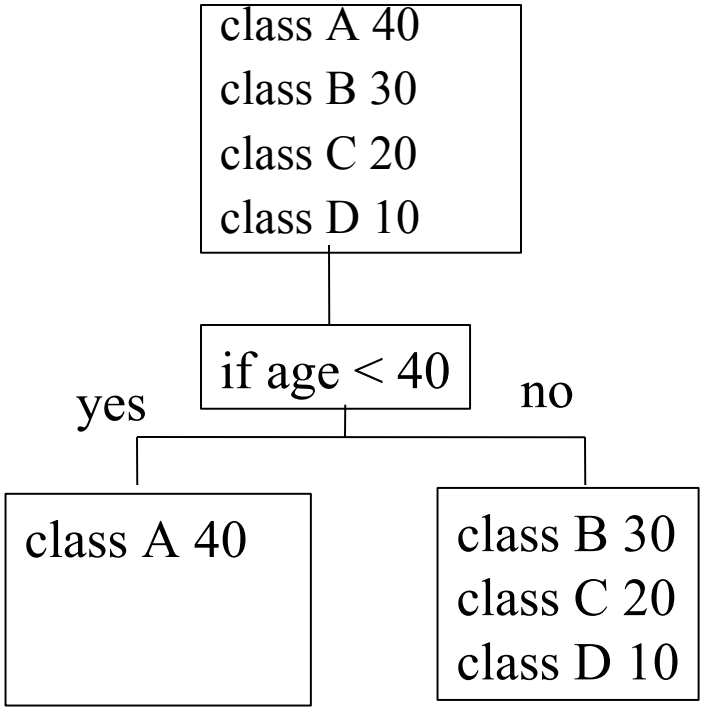


Entropía vs. Indice Gini

- 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.

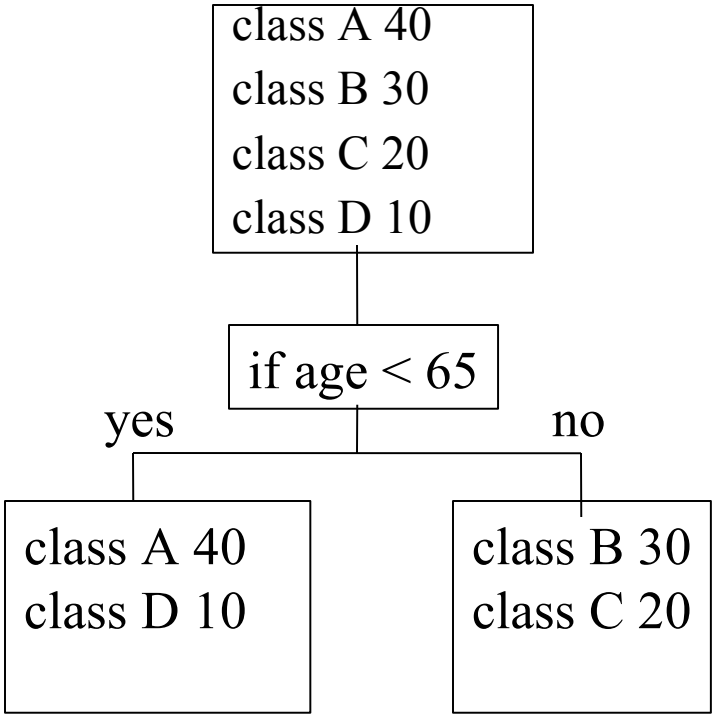
Entropía vs. Indice Gini

- 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