

Arboles de Decisión (II)

Carlos Hurtado L.

Depto de Ciencias de la Computación,
Universidad de Chile

¿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:

- Índice Gini
- Entropía (Ganancia de información)
- Test Chi-cuadrado
- Proporción de Ganancia de Información

Teoría de Información

- Supongamos que transmitimos en un canal (o codificamos) mensajes que son símbolos en $\{x_1, x_2, \dots, x_n\}$.
- A primera vista, necesitamos $\log n$ bits por mensaje.
- Si el emisor y el receptor conocen la distribución de prob. $P(X=x_i)=p_i$, podríamos elaborar un cdg que requiera menos bits por mensaje.

Códigos de Huffman

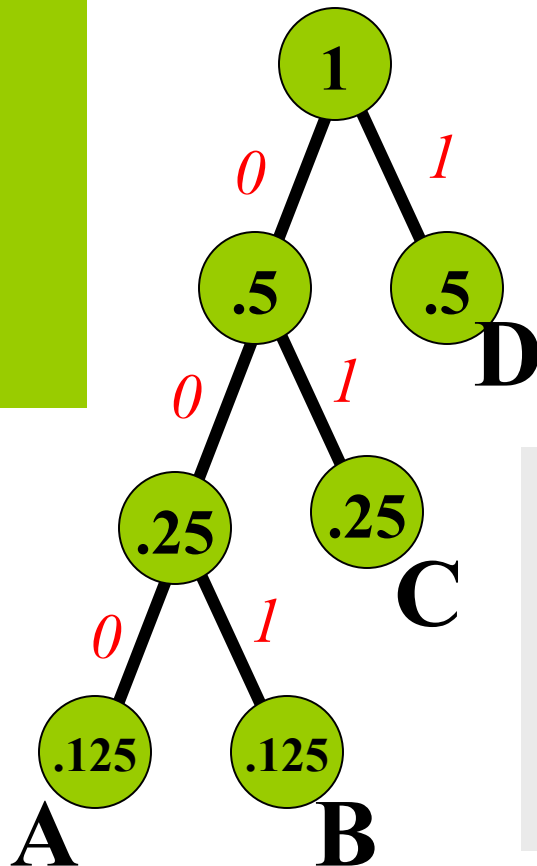
Símb. Prob.

A .125

B .125

C .25

D .5



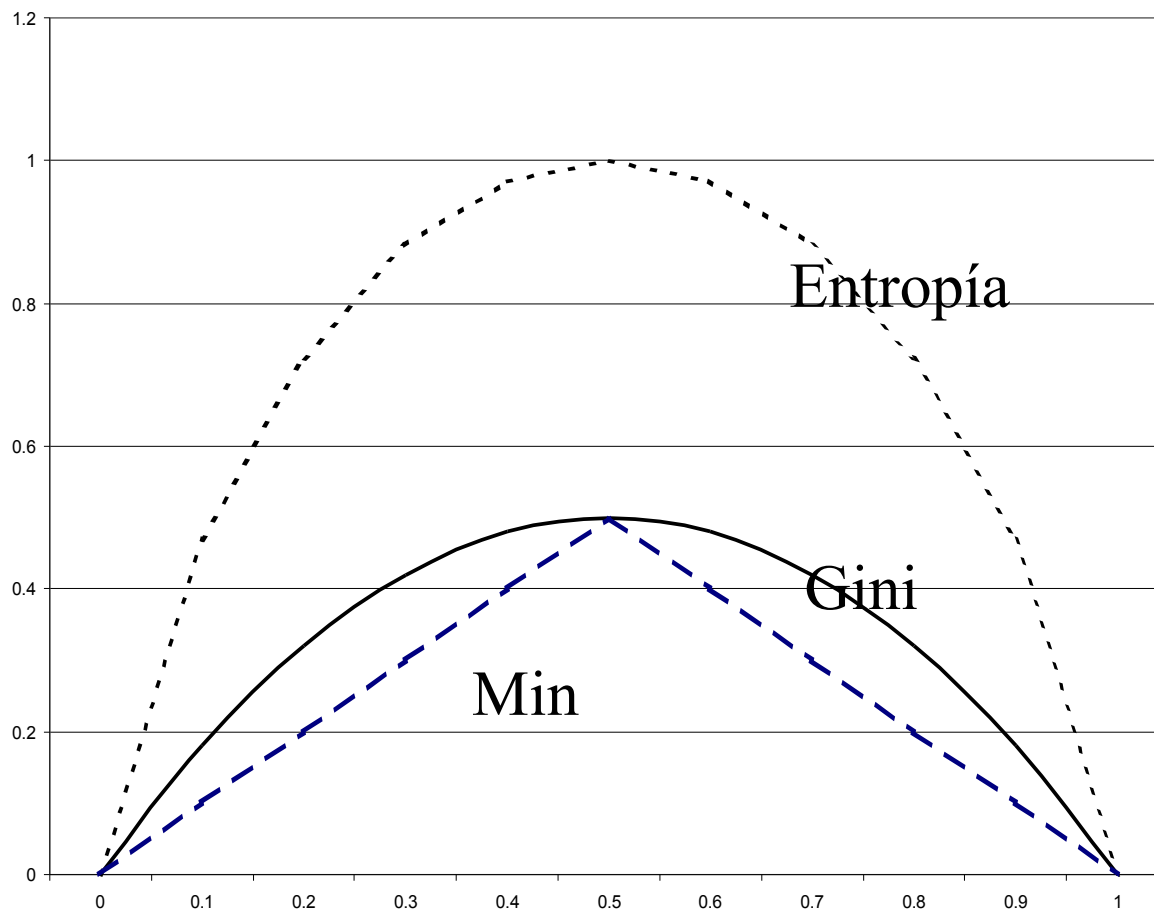
M	cod.	long.	prob	
A	000	3	0,125	0,375
B	001	3	0,125	0,375
C	01	2	0,250	0,500
D	1	1	0,500	0,500
average message length				1,750

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 transmitir un conjunto de mensajes sobre $\{x_1, x_2, \dots, x_n\}$ con distribución de prob. $P(X=x_i)=p_i$
- $\text{Entropy}(P) = -(p_1 \cdot \log(p_1) + p_2 \cdot \log(p_2) + \dots + p_n \cdot \log(p_n))$
- Información asociada a la distribución de probabilidades P .
- Ejemplos:
 - Si P es $(0.5, 0.5)$, $\text{Entropy}(P) = 1$
 - Si P es $(0.67, 0.33)$, $\text{Entropy}(P) = 0.92$
 - Si P is $(1, 0)$, $\text{Entropy}(P)=0$
- Mientras más uniforme es P , mayor es su entropía

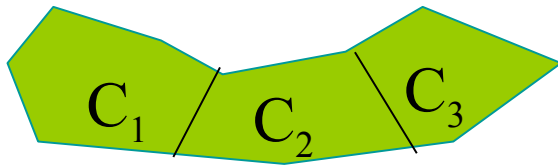
Entropía



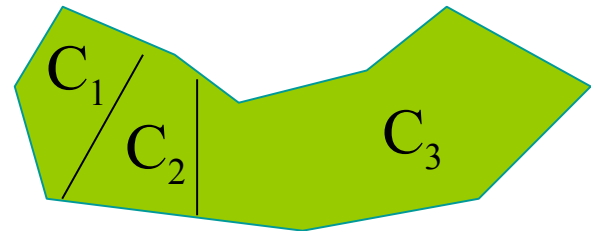
Selección de splits usando Entropía

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

- 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: $0 \cdot \log 0 = 0$)
 - Todas las clases son igualmente frecuentes



Entropía alta



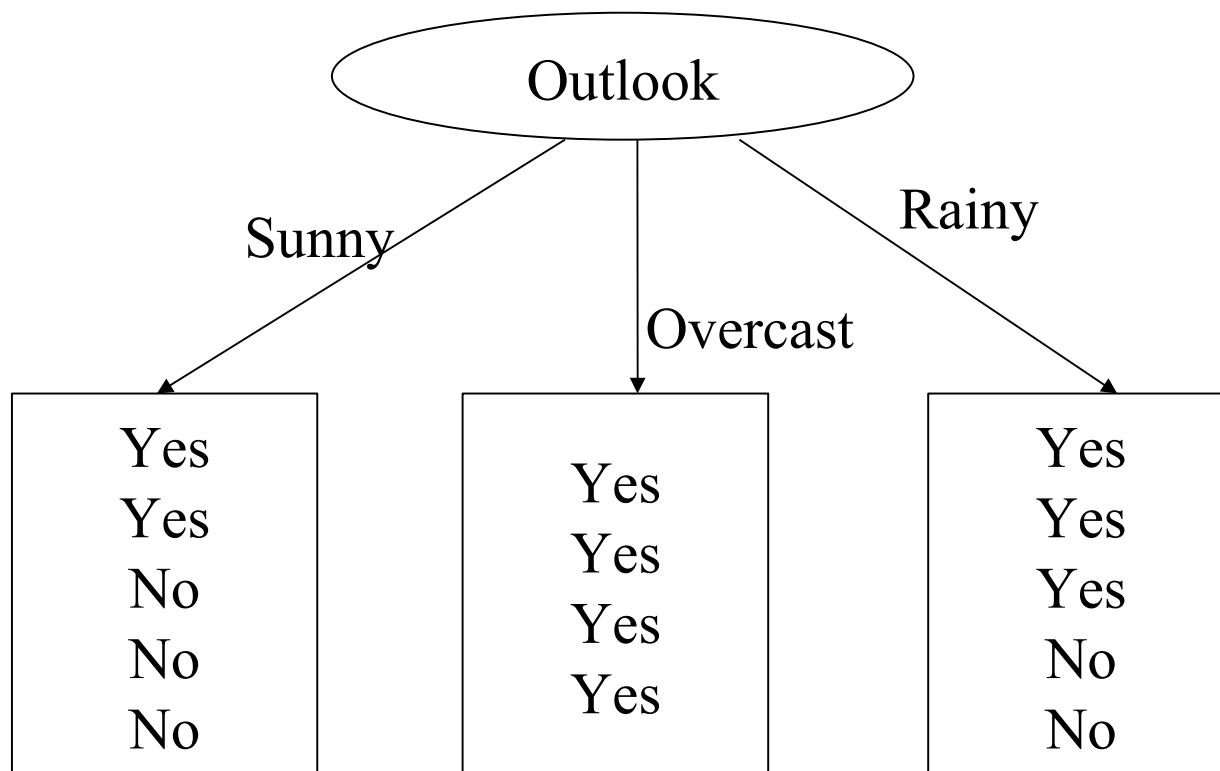
Entropía baja

Selección de Splits: Ganancia de Información

- Criterio para elegir un split: seleccionar el split con la mayor ganancia de información (Gain)
- Equivalente a seleccionar el split con la menor entropía ponderada
- Dado un split $S=\{s_1,\dots,s_n\}$ 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 simple 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

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

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

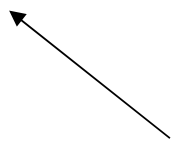
$$= 0.92 - 0.69 = 0.23$$

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

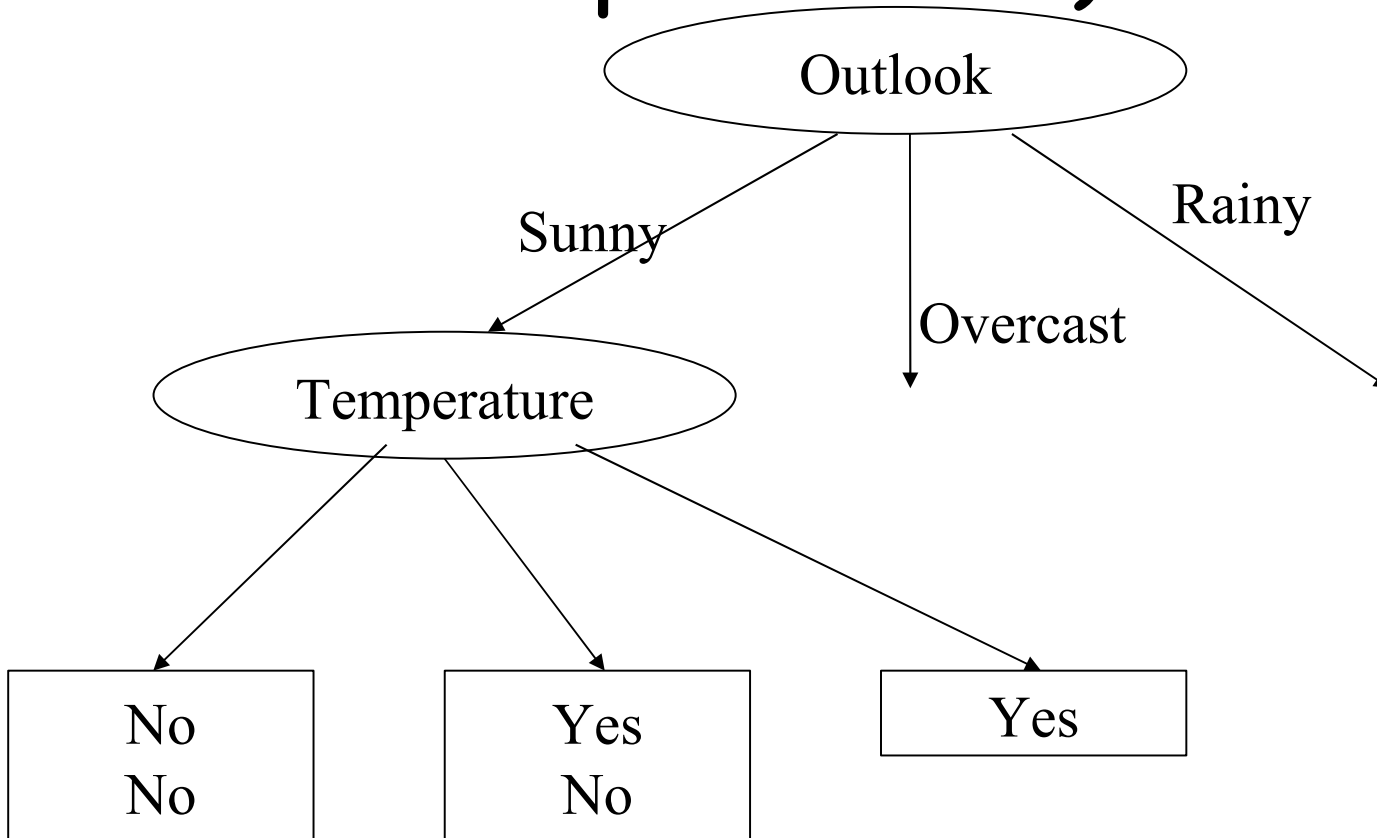
$$Gain(S, humidity) = 0.15$$

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

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"

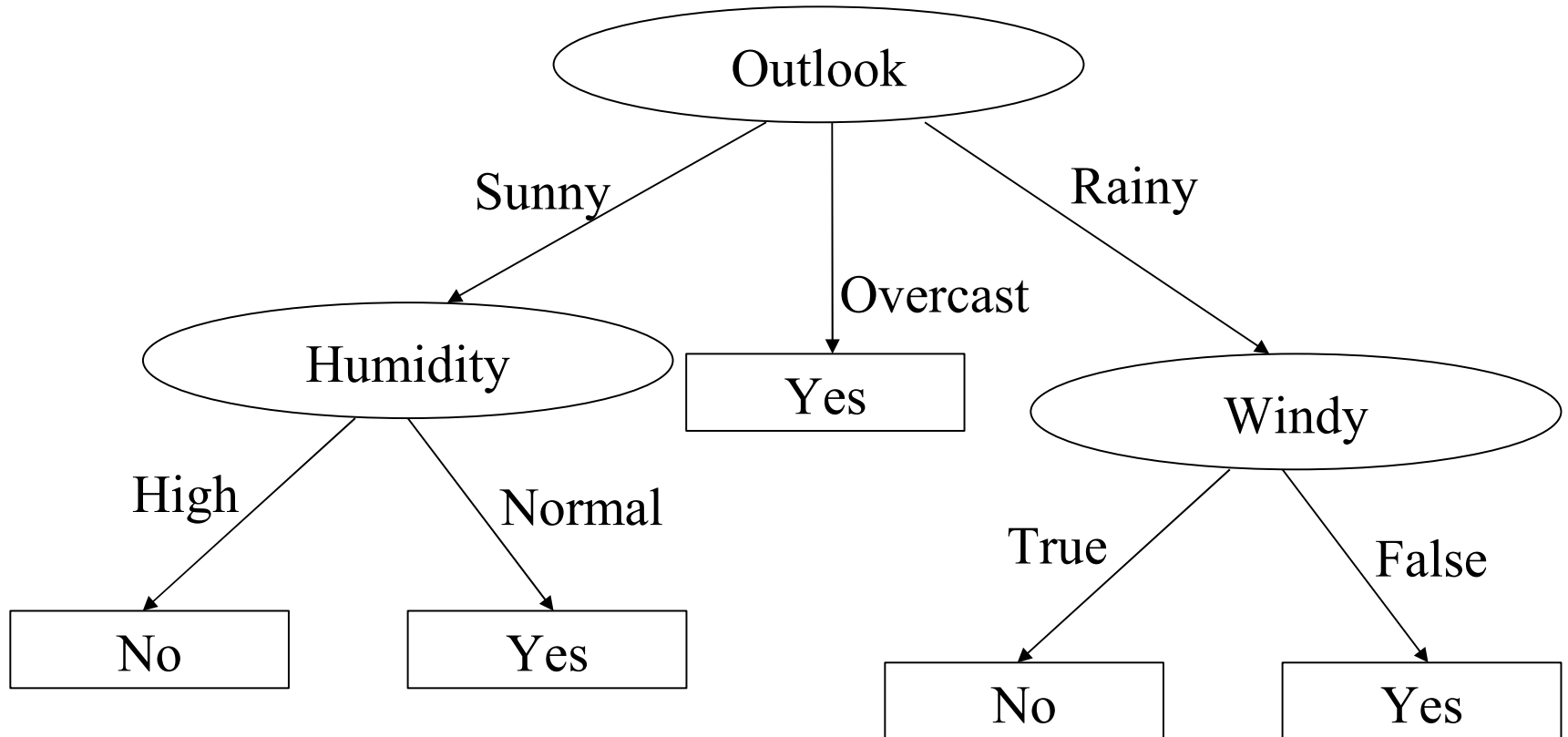
$$\begin{aligned} \text{Gain}(S, \text{temp}) &= \text{Entropy}(S) - \sum_{v \in \text{Values}(a)} \frac{|S_v|}{|S|} \text{Entropy}(S_v) \\ &= 0.97 - \frac{2}{5} \cdot 0 - \frac{2}{5} \cdot 1 - \frac{1}{5} \cdot 0 \\ &= 0.97 - 0.40 = 0.57 \end{aligned}$$

$$\text{Gain}(S, \text{humidity}) = 0.97$$

$$\text{Gain}(S, \text{windy}) = 0.02$$

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

Ejemplo: Arbol resultante

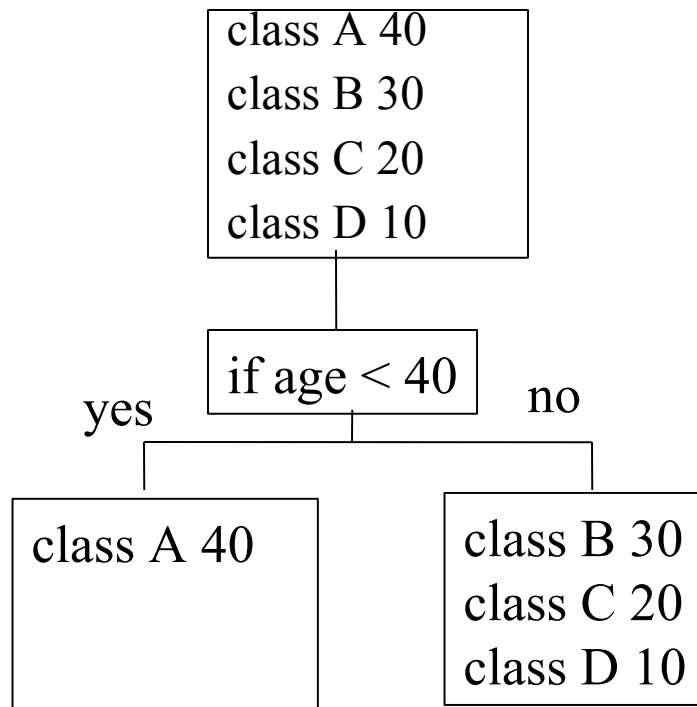


Entropía vs. Índice Gini

- 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.

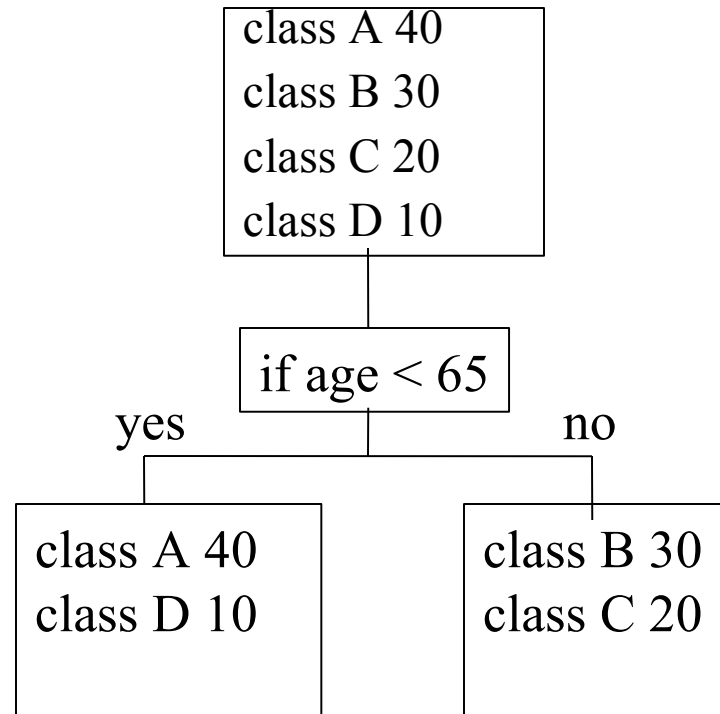
Entropía vs. Índice Gini

- Índice 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

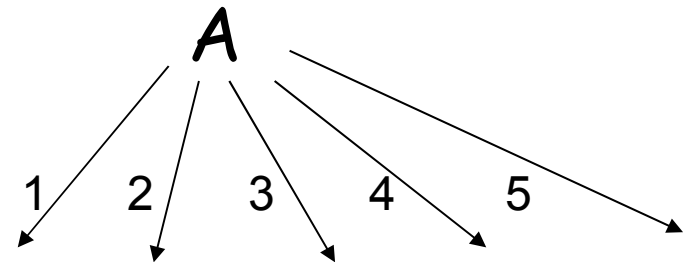


GiniSplit=0,4 EntPond=0,254

Problemas de entropía e Índice Gini

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

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

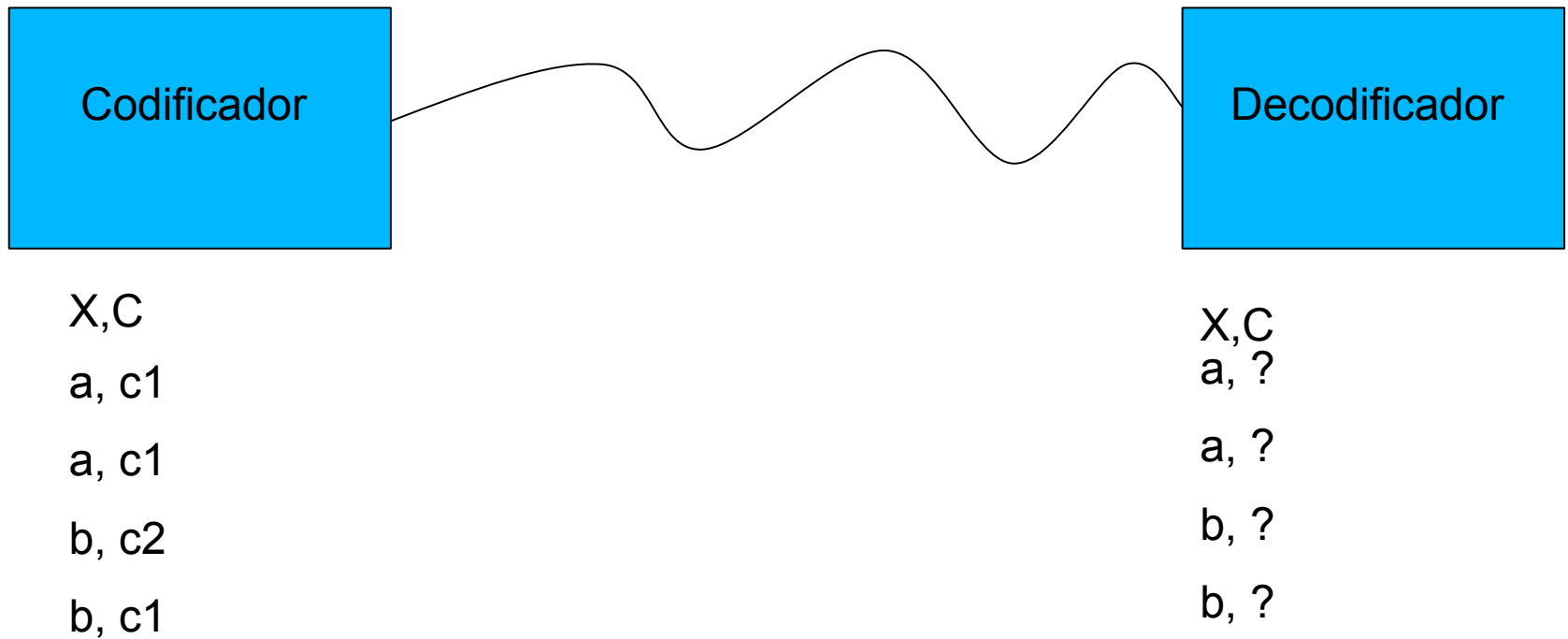


Problemas de entropía e Índice Gini

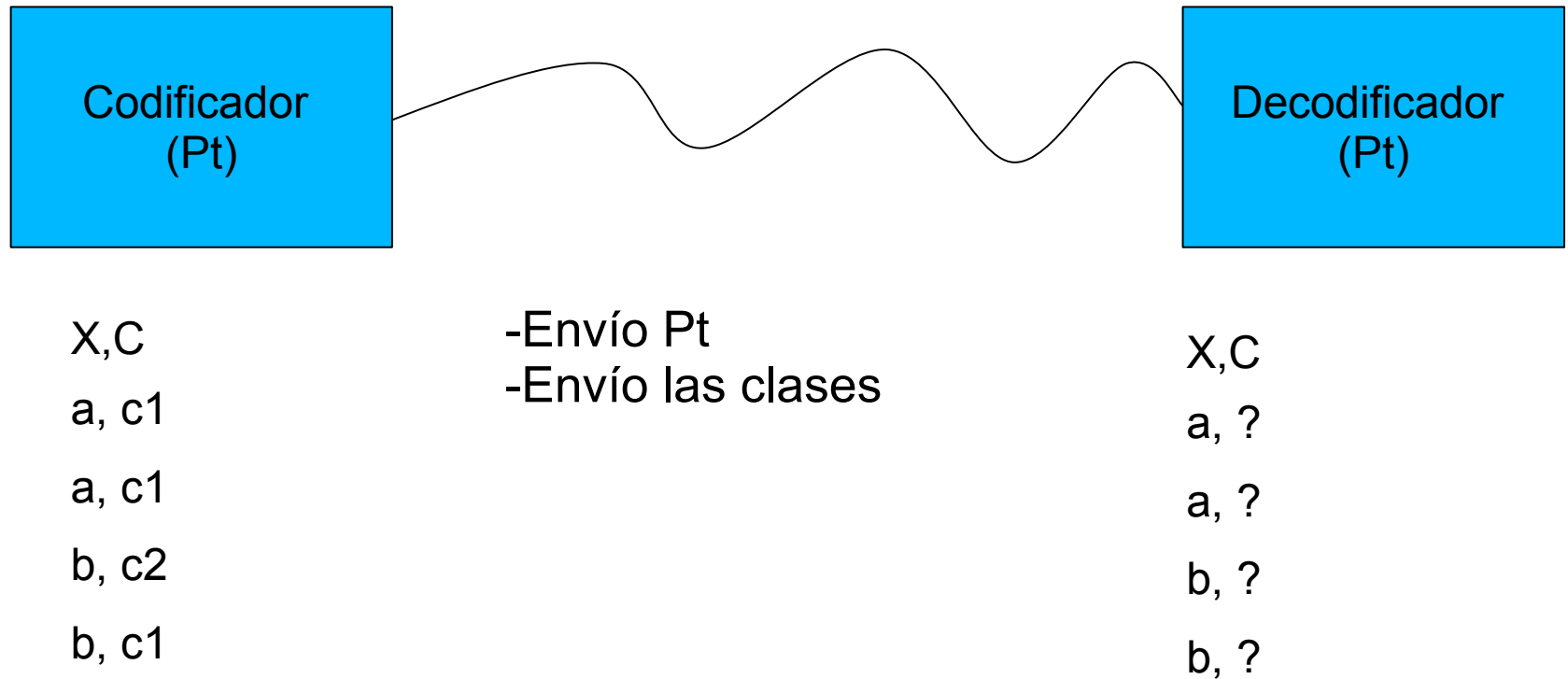
- Favorecen splits con muchos nodos hijos.
Favorece atributos con muchos valores.
- Solución 1.: usar splits binarios
- Solución 2: usar GainRatio

Arboles y Teoría de Información

- Teoría de información nos entrega un marco para medir la calidad de un árbol.
- Idea: usamos el árbol para construir un código eficiente para enviar las clases de los datos. Un buen árbol permite un código más eficiente.



Teoría de la Información: Arbol de un nodo (t)



Cantidad de bits a enviar = $Nt \text{ Entropy}(Pt)$ + costo enviar Pt

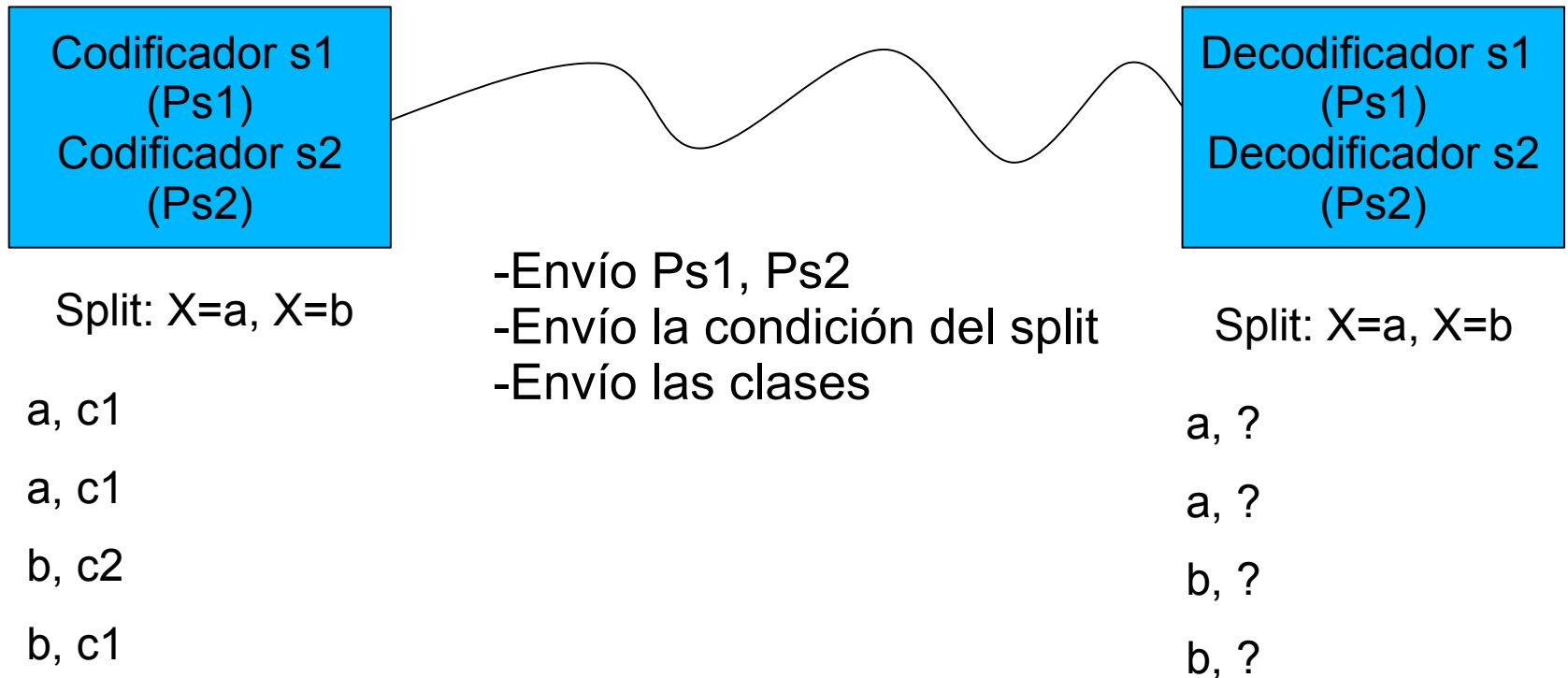
Costo de Enviar P_t

- Si tenemos m clases y N_t datos en el nodo t .
- El número posible de distribuciones de probabilidad es igual al número de soluciones enteras a la ecuación:

$$X_1 + X_2 + \dots + X_n = N_t$$

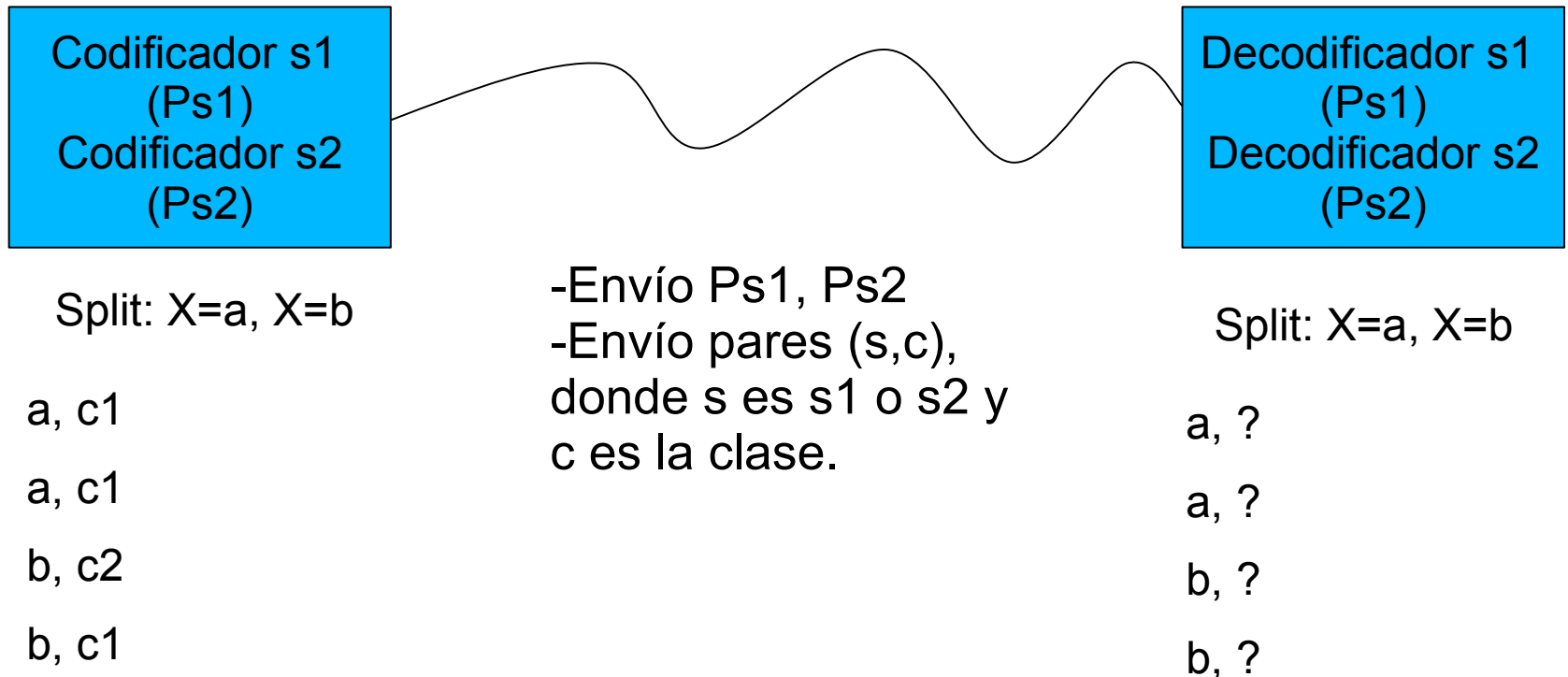
- Esto es $G = N_t + m - 1$ "sobre" $m - 1$.
- Sea G ese número, luego necesitamos $\log(G)$ bits para enviar P_t .

Teoría de la Información: Arbol con un split (t, s1, s2), caso 1



Cantidad de bits a enviar = $Ns1 \text{ Entropy}(Ps1) + Ns2 \text{ Entropy}(s2) + \text{costo de enviar } Ps1, Ps2 + \text{costo de enviar split}$

Teoría de la Información: Arbol con un split (t, s1, s2), caso 2



Cantidad de bits a enviar = $Ns1 \text{ Entropy}(Ps1) + Ns2 \text{ Entropy}(s2) +$
costo de enviar Ps1, Ps2 + costo de enviar el nodo hijo al cuál pertenece
el dato.

Gain Ratio

- Primero calculamos la cantidad de bits necesarios para enviar el nodo hijo al cuál pertenece el dato.

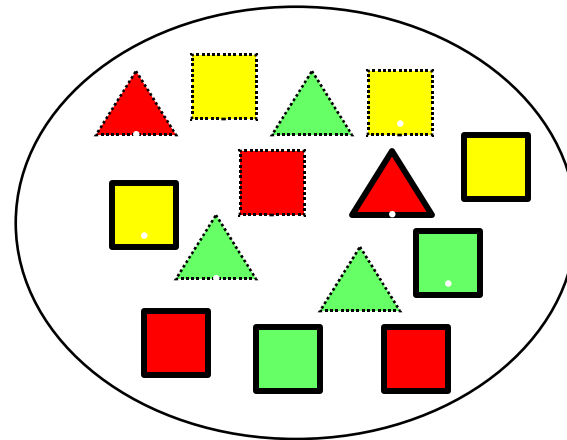
$$SplitInfo(t, S) = Entropy\left(\frac{|S_1|}{|t|}, \frac{|S_2|}{|t|}, \dots, \frac{|S_n|}{|t|}\right)$$

- Luego definimos el "Gain Ratio" como:

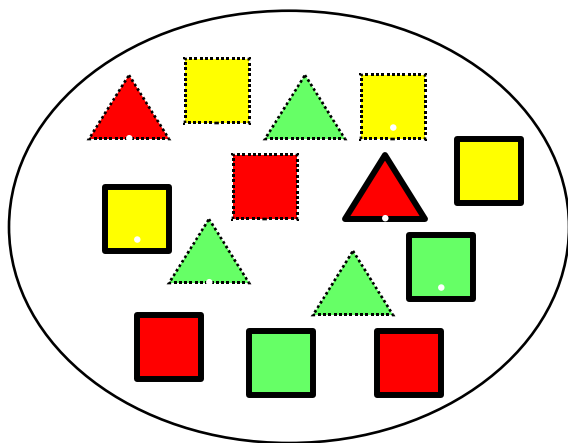
$$GainRatio(t, S) = Gain(t, S) / SplitInfo(t, S)$$

Ejemplo: Triángulos y Cuadrados

#	Attribute			Shape
	Color	Outline	Dot	
1	green	dashed	no	triange
2	green	dashed	yes	triange
3	yellow	dashed	no	square
4	red	dashed	no	square
5	red	solid	no	square
6	red	solid	yes	triange
7	green	solid	no	square
8	green	dashed	no	triange
9	yellow	solid	yes	square
10	red	solid	no	square
11	green	solid	yes	square
12	yellow	dashed	yes	square
13	yellow	solid	no	square
14	red	dashed	yes	triange



Entropía



- 5 triangles
- 9 squares
- class probabilities

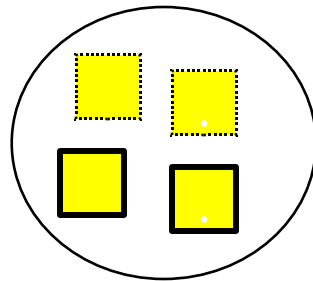
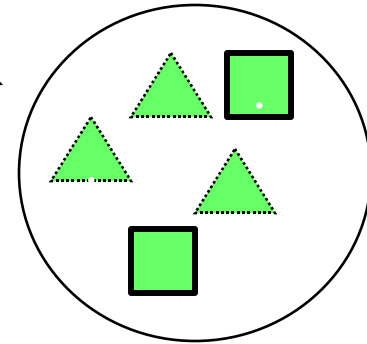
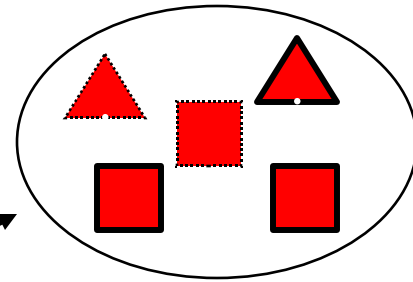
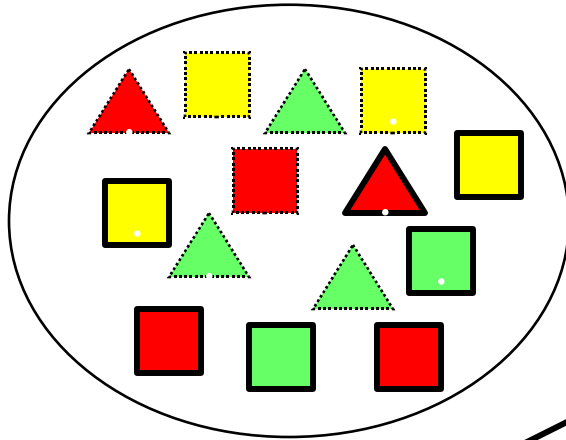
$$p(\square) = \frac{9}{14}$$

$$p(\triangle) = \frac{5}{14}$$

- entropy

$$I = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0.940 \text{ bits}$$

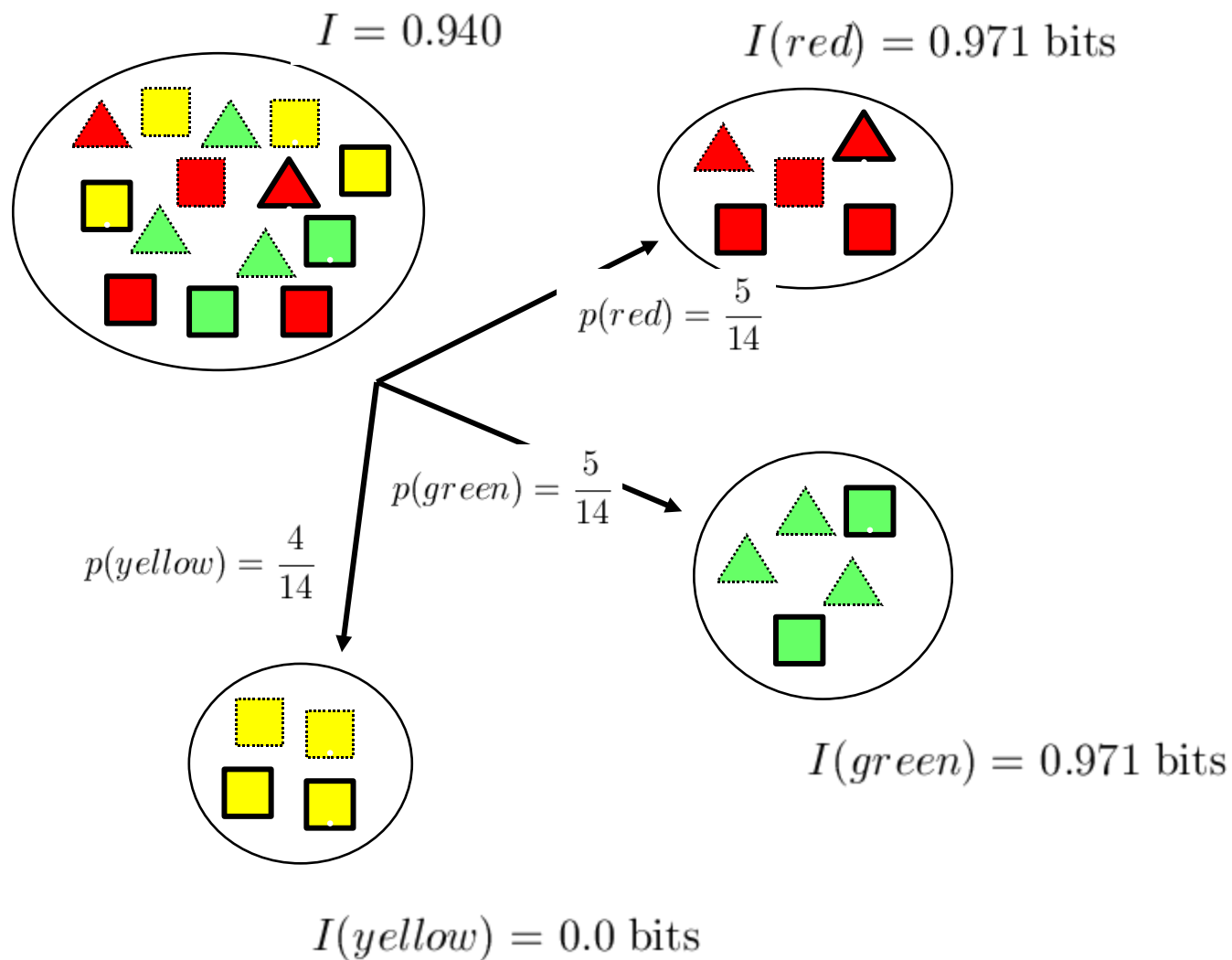
$$I(\text{red}) = -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} = 0.971 \text{ bits}$$



$$I(\text{green}) = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0.971 \text{ bits}$$

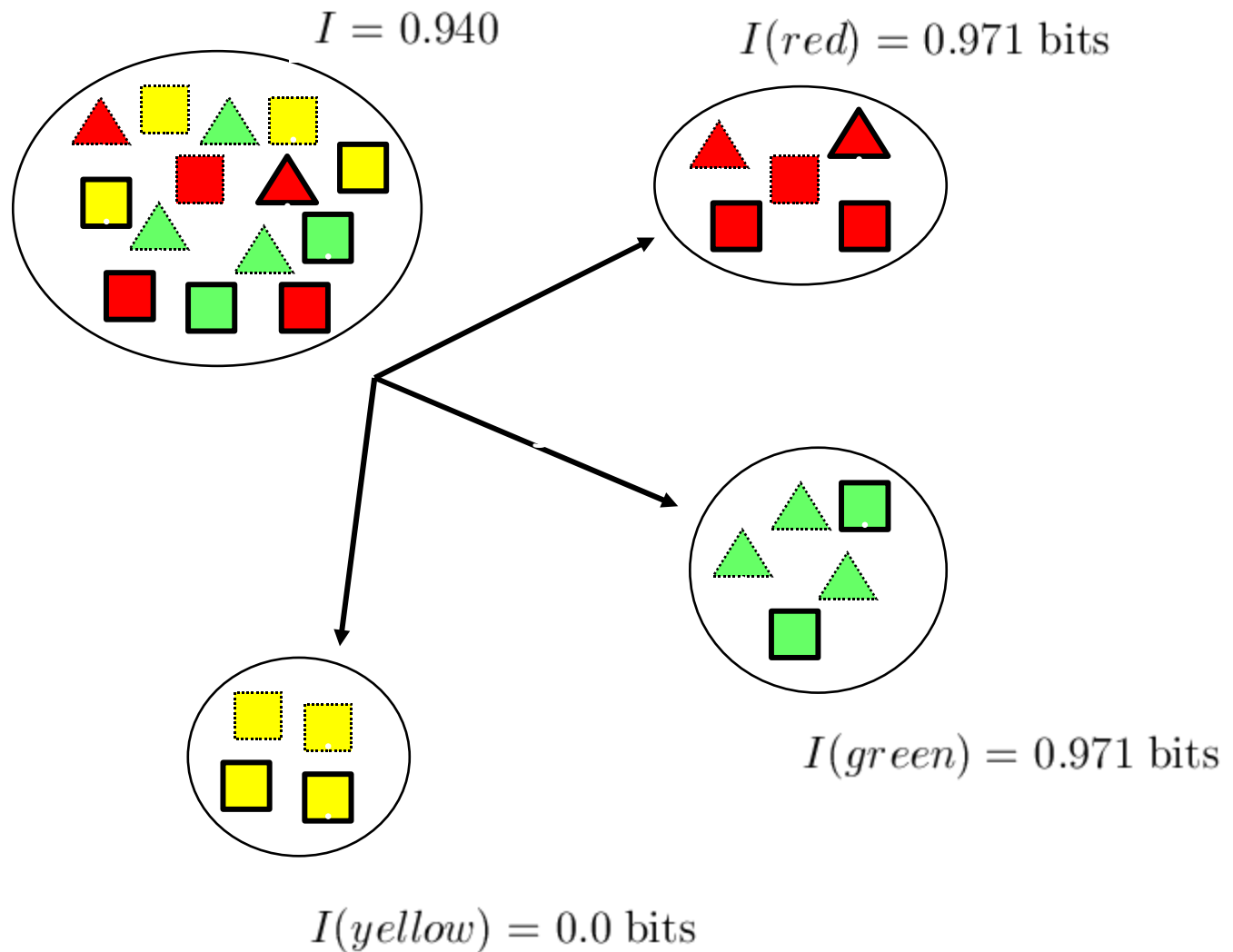
$$I(\text{yellow}) = -\frac{4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2 \frac{0}{4} = 0.0 \text{ bits}$$

Reducción
de la
Entropía por
un Split

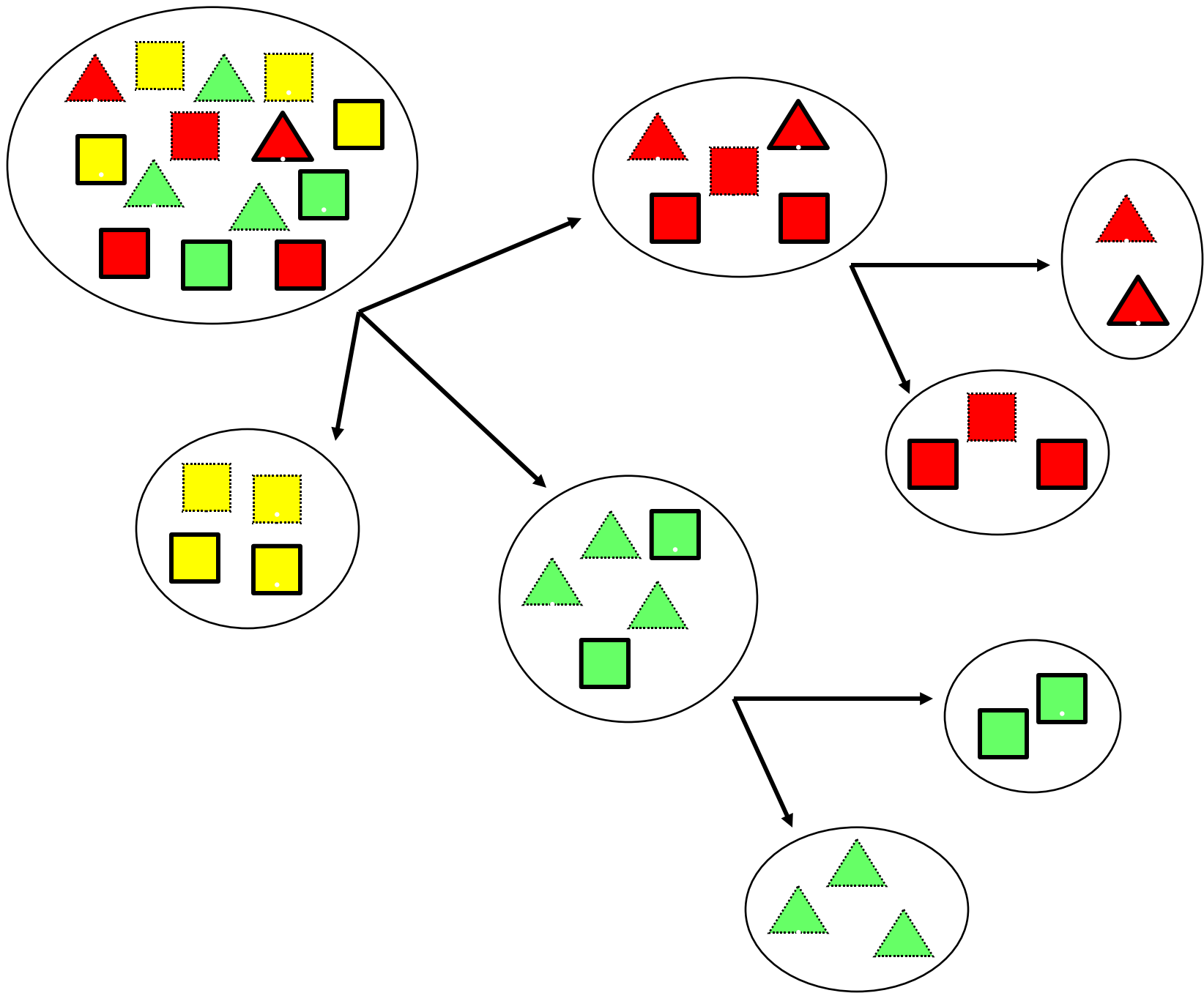


$$I_{res}(\text{Color}) = \sum p(v)I(v) = \frac{5}{14}0.971 + \frac{5}{14}0.971 + \frac{4}{14}0.0 = 0.694 \text{ bits}$$

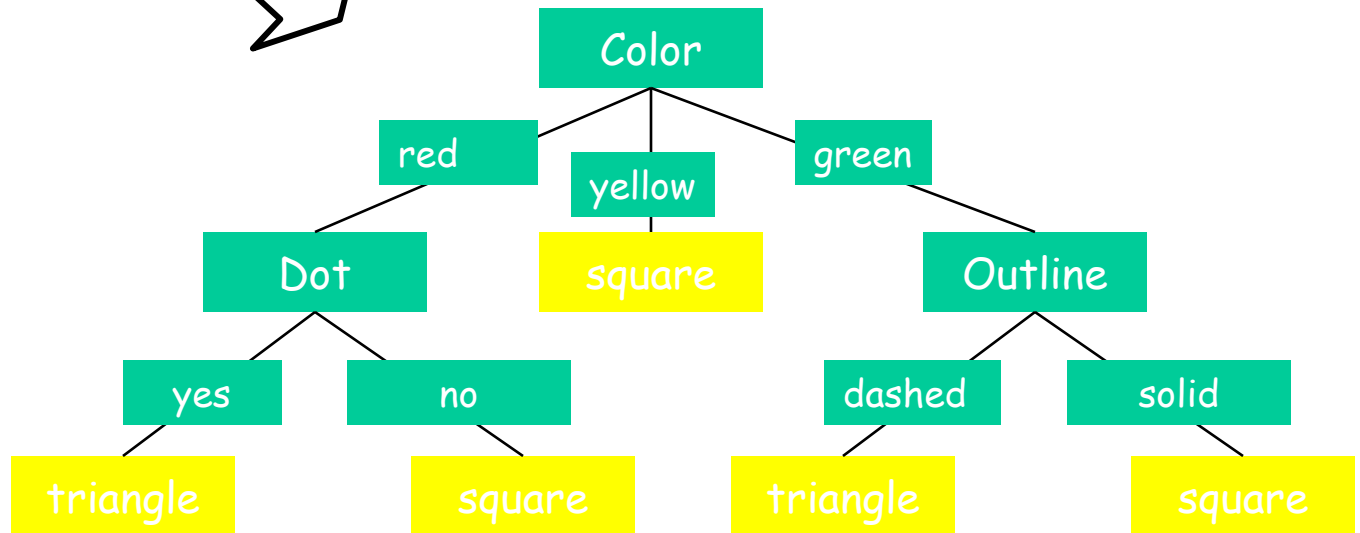
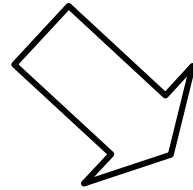
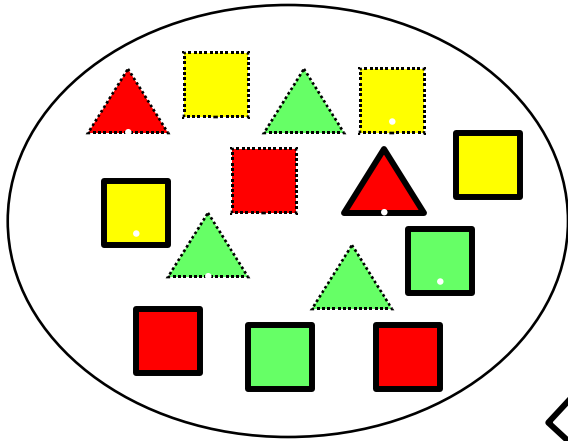
Ganancia de Información



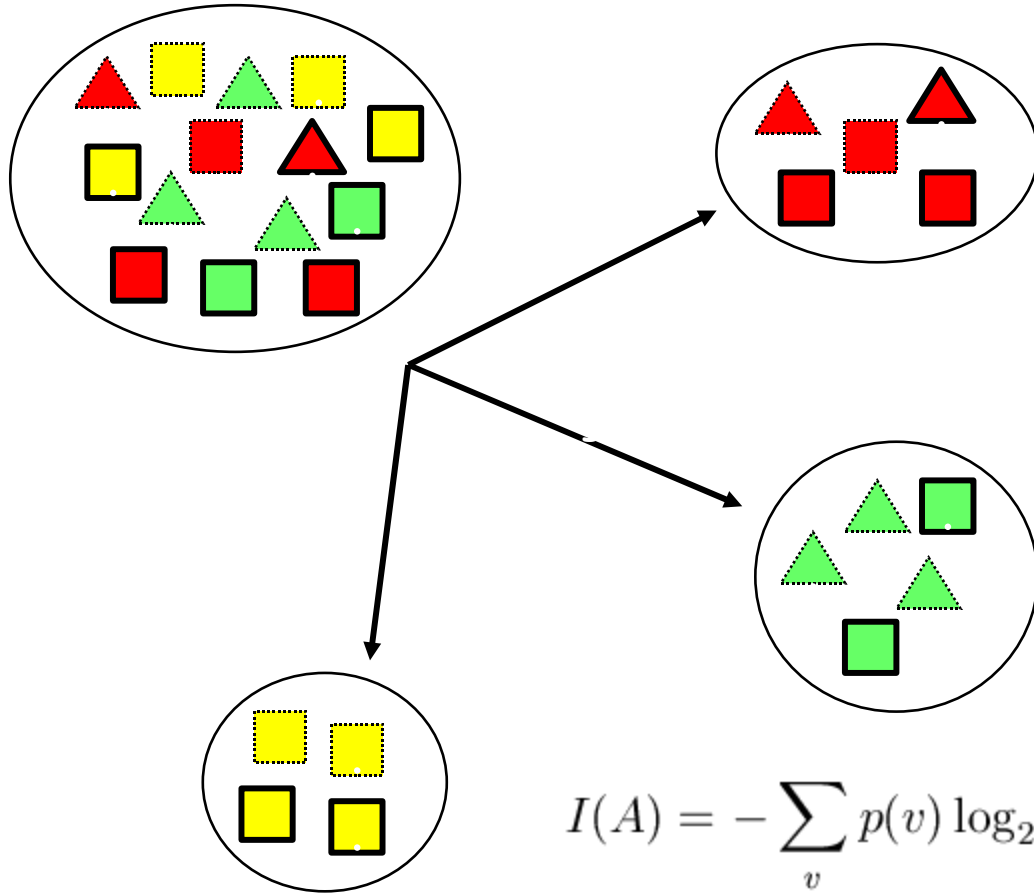
$$Gain(\text{Color}) = I - I_{res}(\text{Color}) = 0.940 - 0.694 = 0.246 \text{ bits}$$



Arbol Resultante



Information Gain Ratio



$$I(A) = - \sum_v p(v) \log_2(p(v))$$

$$I(\text{Color}) = -\frac{5}{14} \log_2 \frac{5}{14} - \frac{5}{14} \log_2 \frac{5}{14} - \frac{4}{14} \log_2 \frac{4}{14} = 1.58 \text{ bits}$$

$$\text{GainRatio}(\text{Color}) = \frac{\text{Gain}(\text{Color})}{I(\text{Color})} = \frac{0.940 - 0.694}{1.58} = 0.156$$

Information Gain and Information Gain Ratio

A	$ v(A) $	$Gain(A)$	$GainRatio(A)$
Color	3	0.247	0.156
Outline	2	0.152	0.152
Dot	2	0.048	0.049

Criterios de Selección para el Ejemplo

<i>A</i>	<i>Gain(A)</i>	<i>GainRatio(A)</i>	<i>GiniGain(A)</i>
Color	0.247	0.156	0.058
Outline	0.152	0.152	0.046
Dot	0.048	0.049	0.015

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.

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.
 - Tenemos dos variables Café (C) y Té (T)
 - Valores son compró o no-compró

Tabla de Contingencia

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

Tabla de Contingencia (cont.)

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

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

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

Ejemplo: valor esperado E

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

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

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

Estimador Chi-cuadrado

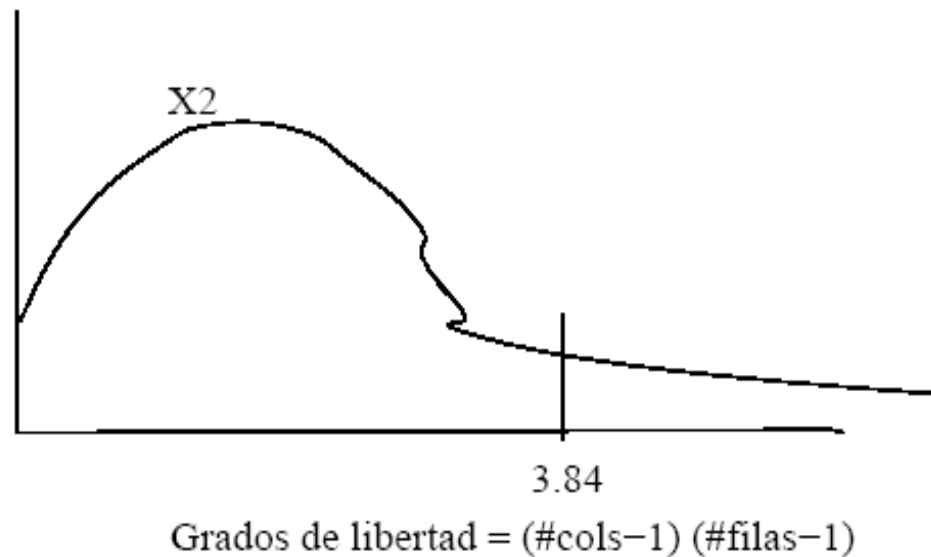
Estimador Chi-Cuadrado:

$$\chi^2 = \sum_{r \in R} \frac{(O(r) - E[r])^2}{E[r]}$$

Desviación normalizada entre la tabla de contingencia real
y la esperada

Estimador Chi-cuadrado (cont.)

Si χ^2 es mayor que 3.84 rechazamos la suposición de independencia con un 95% de nivel de confianza.

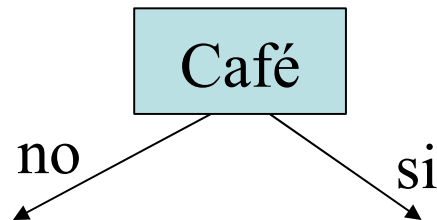


Uso de Chi-cuadrado para seleccionar splits

- Consideremos 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.

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

Medidas de impureza:

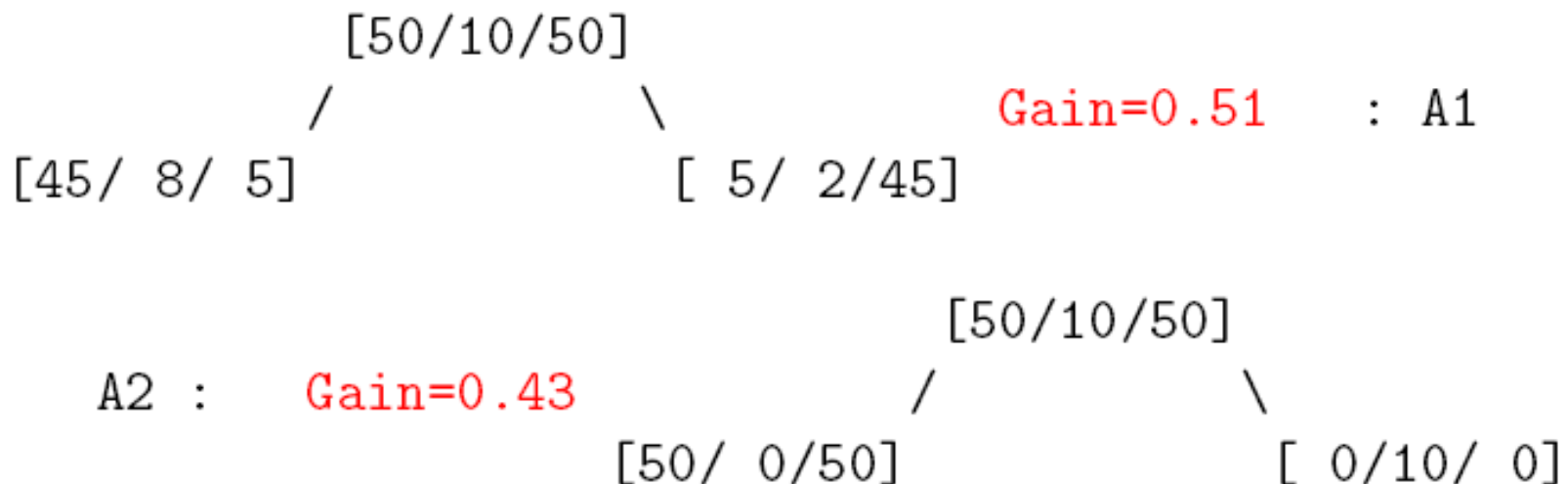
Estudio de Mingers (1989)

- Los árboles obtenidos con distintos 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:



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

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

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

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.