Métodos Basados en Casos y en Vecindad

CC52A - Inteligencia Artificial

Gonzalo Ríos D.

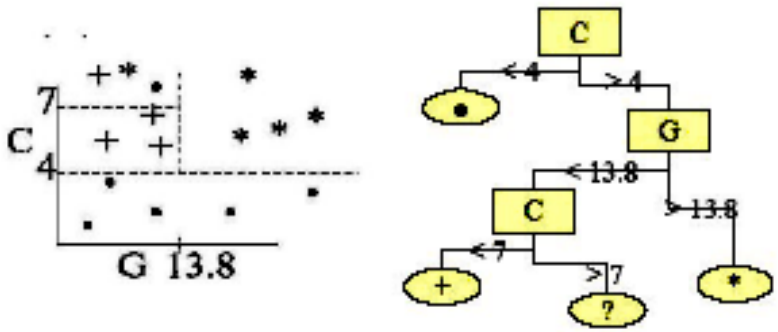
DCC - UChile

Otoño 2011

Arboles de Decisión

Definiciones Básicas

- Los árboles de decisión son modelos en grafos, particularmente árboles.
- Los nodos internos representan variables, o un conjunto de variables
- Los arcos son proposiciones booleanas sobre esas variables.
- Las hojas representan la respuesta, la clase o la propiedad buscada sobre el dato.

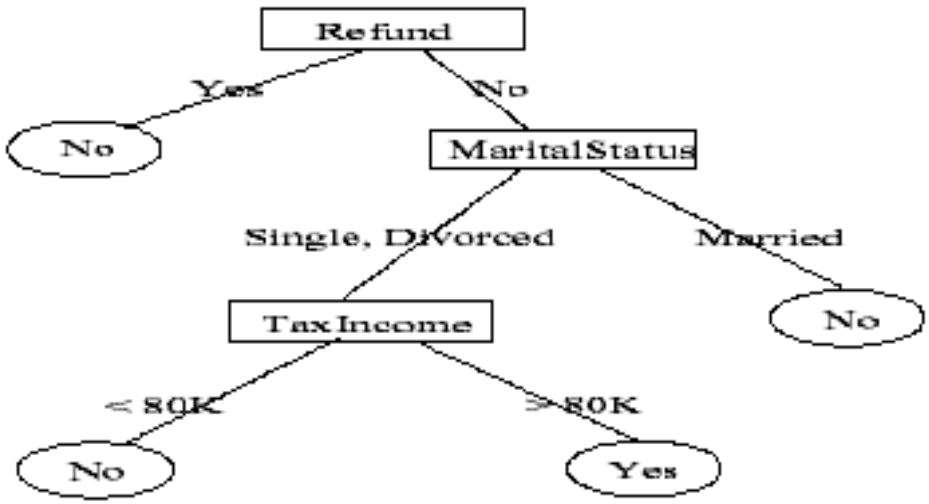


Arboles de Decisión

Definiciones Básicas

Los árboles de decisión tienen muy buenas propiedades, entre ellas:

- Fáciles de construir
- Fáciles de interpretar
  - No sólo sirven para clasificar nuevos datos, sino que para qué determina cada clase.
  - Modelan la frontera de decisión
- Buena precisión en muchos casos



Definición

Un split es una variable más una lista de condiciones sobre la variable.
Ejemplo: $(A,\{a_1, ..., a_n\})$

- Split categóricos
  - Simples: $\{A=a_1, ..., A = a_n\}$
  - Complejos:$\{A\in \{a_1, ..., a_k\}, A \in \{a_{k+1}, ..., a_n\}\}$
- Split numéricos
  - Estáticos: Discretizando en un inicio
  - Dinámicos
    - Decisión binaria: A<= v, A > v
    - Rangos: [0,15k), [15k,60k), [60k,100k)
    - Combinación lineal de variables

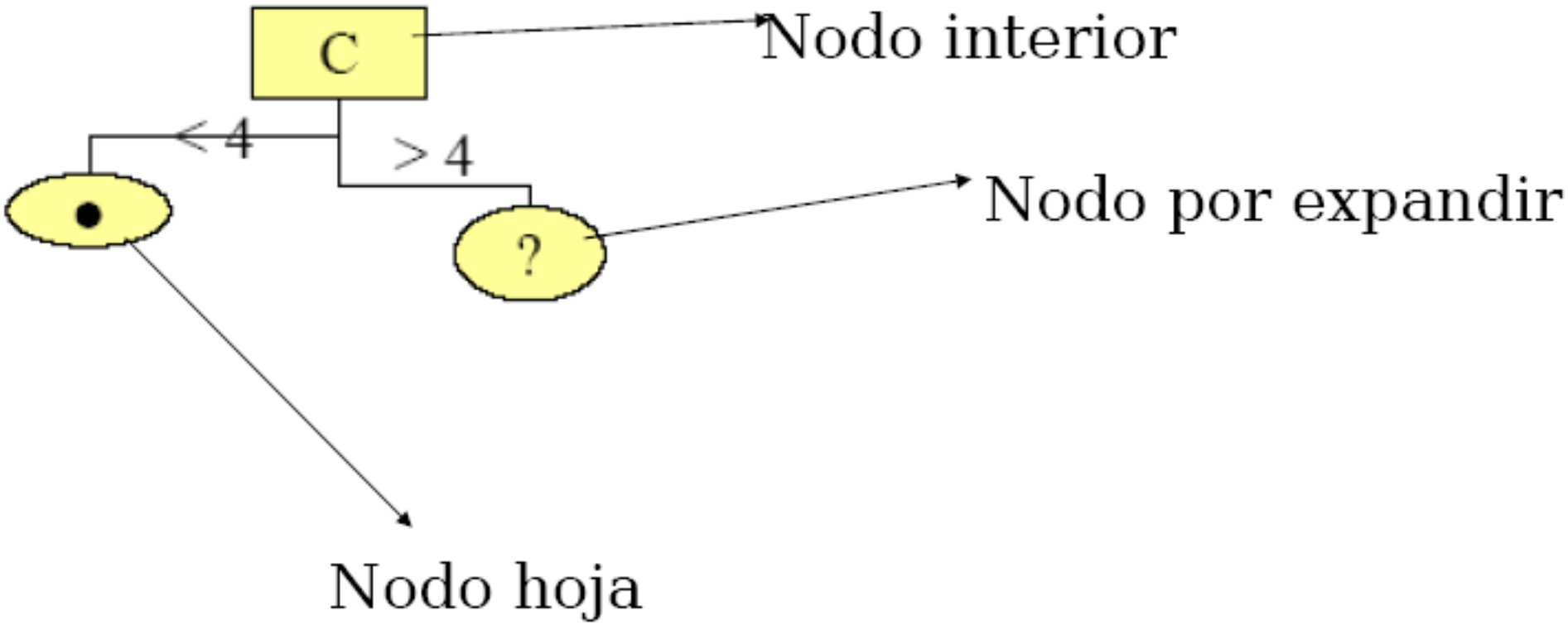
Arboles de Decisión

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

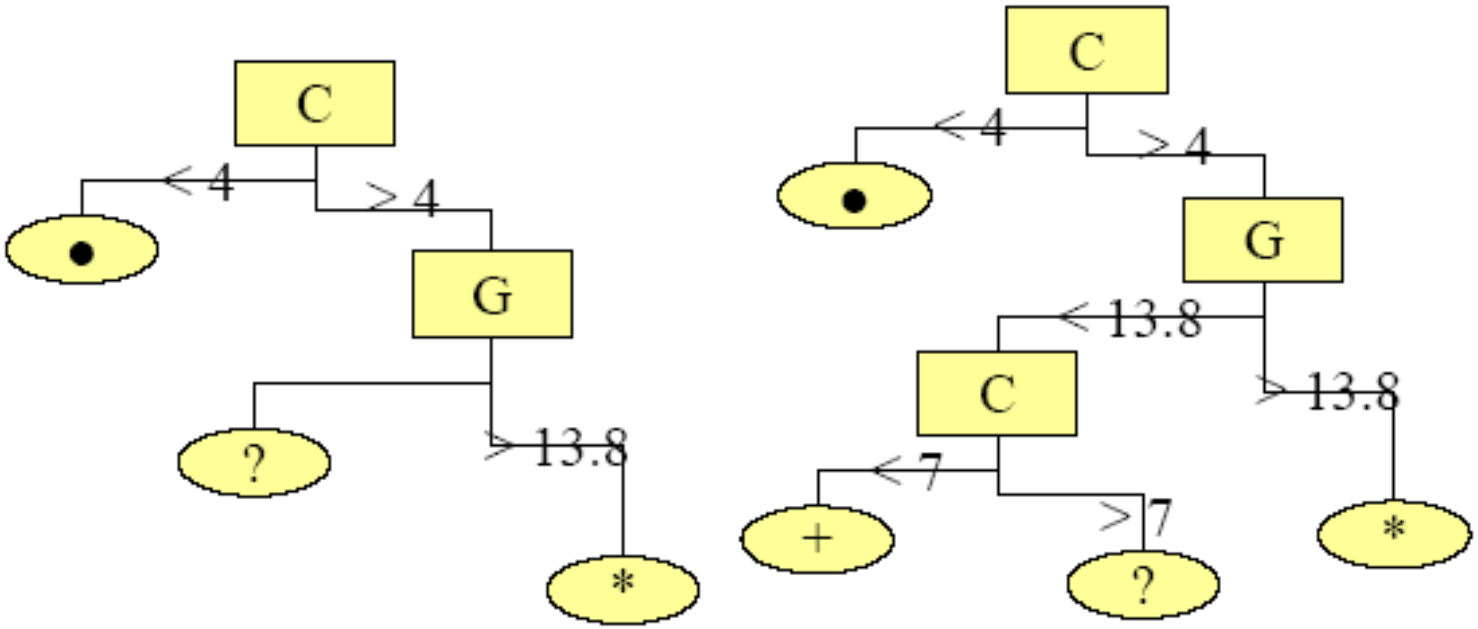
Arboles de Decisión

Algoritmo de Hunt



Arboles de Decisión

Algoritmo de Hunt



Algoritmo de Hunt

- Main (T)
  - Expandir(T)
- Expandir (S)
  - if (Todos los datos están en la misma clase) then return
  - Encontrar el mejor split r
  - Usar r para particionar S en S_1 *y* S_2
  - Expandir (S_1)
  - Expandir (S_2)

Arboles de Decisión

Algoritmo de Hunt

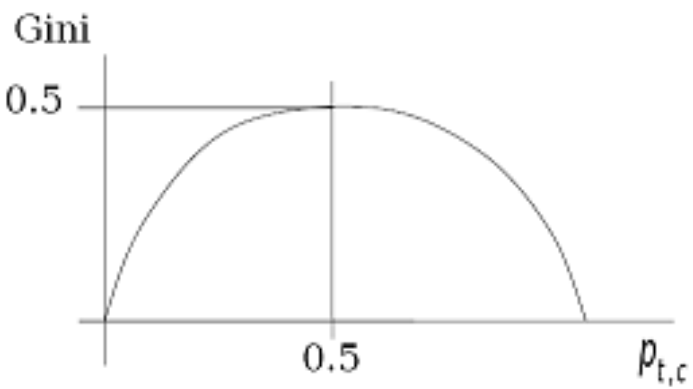
- Las operaciones de expansión se realizan "primero en profundidad"
- Lo complejo es encontrar el mejor split en cada operación de expansión
- Número de splits a buscar depende de si el atributo es categórico o no y del tipo de split (e.g., complejo vs. simple).
- Buscamos splits que generen nodos hijos con la menor impureza posible (mayor pureza posible)
- Existen distintos métodos para evaluar splits.
  - Indice Gini
  - Entropía (Ganancia de información)
  - Test Chi-cuadrado
  - Proporción de Ganancia de Información

Definición

Dado un nodo t, se define $Gini(t) = 1 - \sum_{c \in C} p_{t,c}^2$, *donde* $p_{t,c}$ *es la probabilidad de ocurrencia de la clase c en el nodo t.*

- Gini(t): probabilidad de NO sacar dos registros de la misma clase del nodo
- Menor Gini(t) implica mayor pureza

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



Definición

Dado un split $S = \{s_1, .., s_n\}$ *del nodo* t, *se define*

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

- Criterio de selección: selecciónar el split con menor gini ponderado (GiniSplit)
- Veamos un ejemplo para que todo quede más claro

Arboles de Decisión

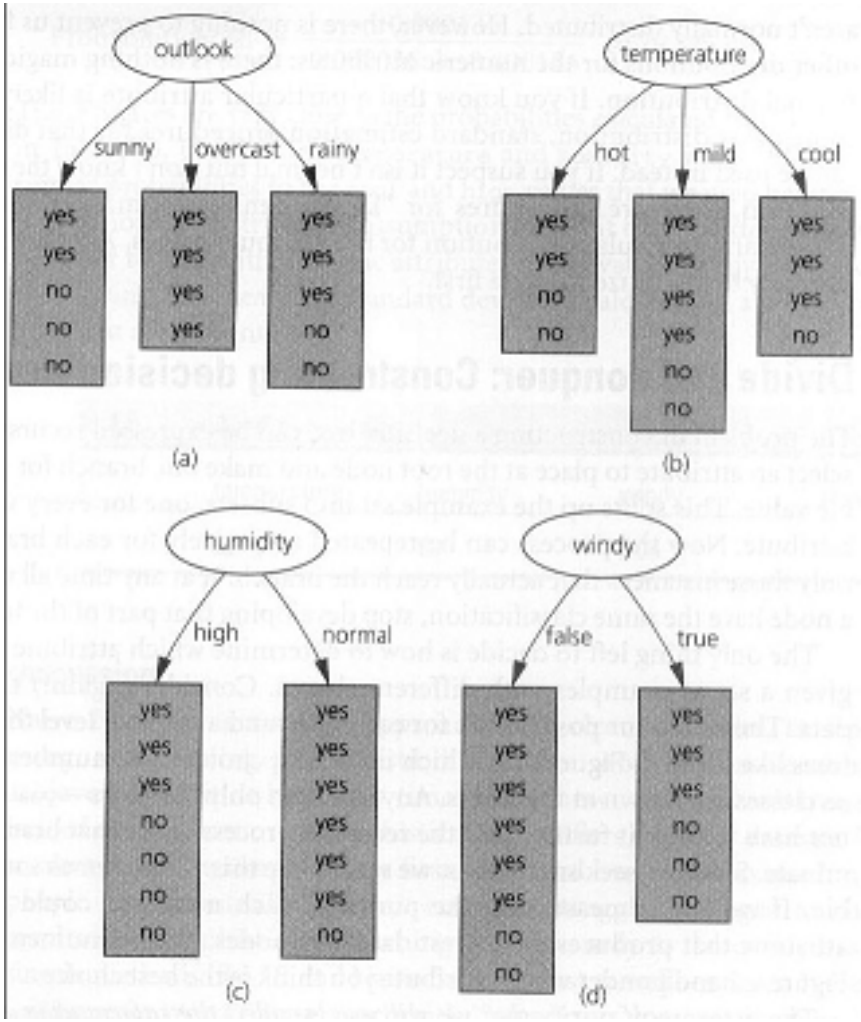
GiniSplit

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Los posibles Splits serían Outlook, Temperature, Humidity y Wind

Arboles de Decisión

GiniSplit



Arboles de Decisión

GiniSplit

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								

- Gini(sunny)=$1 - \left(\frac{2}{5}\right)^2 - \left(\frac{3}{5}\right)^2 = 0.48$
- Gini(overcasr)=$1 - \left(\frac{4}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0$
- Gini(rainy)=$1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = 0.48$
- GiniSplit(Outlook)=$\frac{5}{14} * 0.48 + \frac{4}{15} * 0 + \frac{5}{14} * 0.48 = 0.343$

Arboles de Decisión

Entropía

- De forma análoga, podemos usar la entropía como criterio de selección de split
- $Entropía(t) = -\sum_{c \in C} p_{t,c} \log_2 p_{t,c}$
- La entropía mide la impureza de los datos S, ya que mide la información (número de bits) promedio necesaria para codificar las clases de los datos en el nodo t
- Criterio para elegir un split: selecciónar el split con la mayor ganancia de información (Gain)

Definición

Dado un split $S = \{s_1, .., s_n\}$ *del nodo* t, *se define*

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

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								

- Entropía(Outlook)$=-\frac{9}{14}\log_2\frac{9}{14}-\frac{5}{14}\log_2\frac{5}{14}=0.94$
- Entropía(sunny)$=-\frac{2}{5}\log_2\frac{2}{5}-\frac{3}{5}\log_2\frac{3}{5}=0.97$
- Entropía(overcasr)$=-\frac{4}{4}\log_2\frac{4}{4}-\frac{0}{4}\log_2\frac{0}{4}=0$
- Entropía(rainy)$=-\frac{3}{5}\log_2\frac{3}{5}-\frac{2}{5}\log_2\frac{2}{5}=0.97$
- Gain(Outlook)$=0.94-(\frac{5}{14}*0.97+\frac{4}{15}*0+\frac{5}{14}*0.97)=0.25$

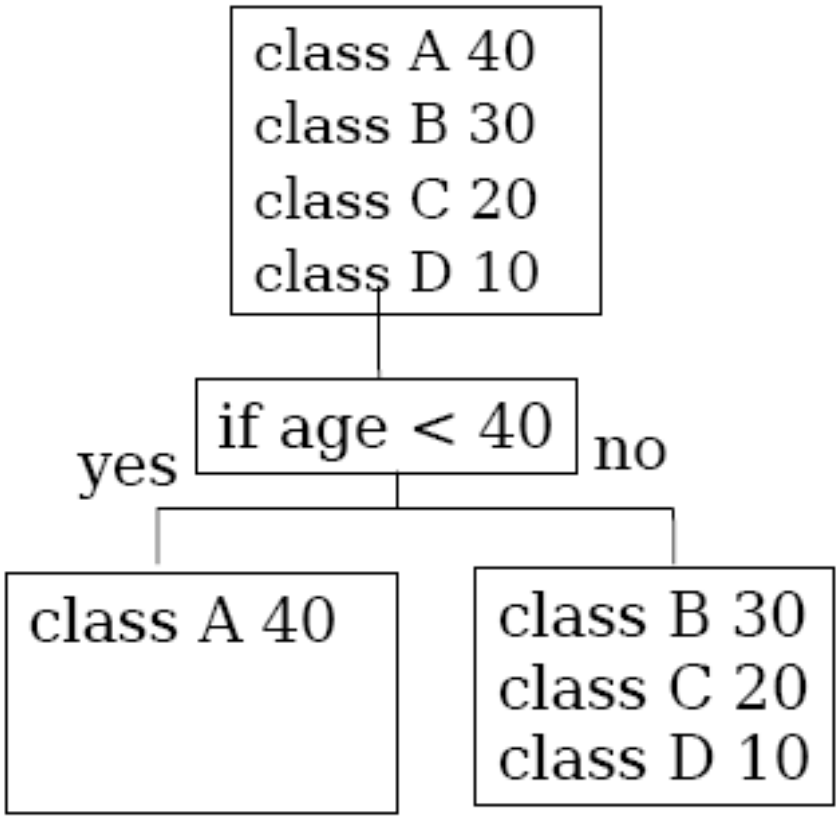
Arboles de Decisión

Gini vs Entropía

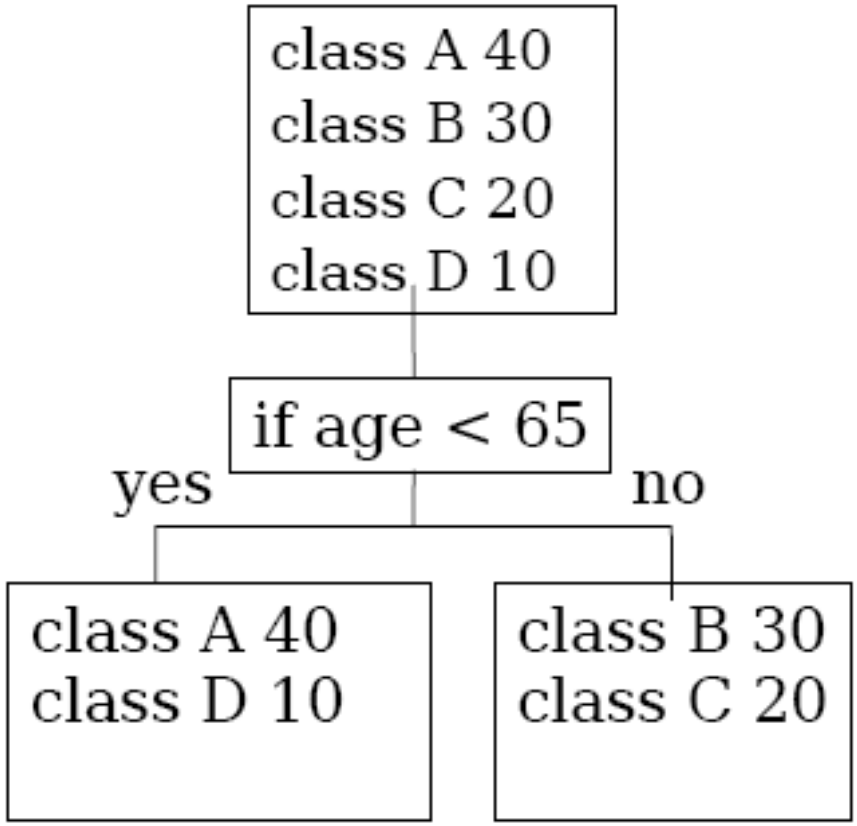
- Gini
  - Tiende a seleccionar splits que aislan una clase mayoritaria en un nodo
  - Tiende a crear splits desbalanceados
  - Tiende a aislar clases numerosas de otras clases
- Entropía
  - Favorece splits balanceados en número de datos
  - Tiende a encontrar grupos de clases que suman más del 50% de los datos

Arboles de Decisión

Gini vs Entropía



GiniSplit = 0,264 EntPond=0,259



GiniSplit=0,4 EntPond=0,254

Arboles de Decisión

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.
- Usando el test de Chi-cuadrado medimos la dependencia entre la variable del split y la variable de la clase

Definición

Tabla de contingencia real: Cada celda $r=(r_1, ..., r_m)$ *contiene el número de veces* $O(r)$ *que ocurren juntos en los datos los valores* $(r_1, ..., r_m)$

Definición

Tabla de contingencia esperada: Para cada celda $r=(r_1, ..., r_m)$ *definimos el valor esperado de la celda si las variables son indep:*

$$E(r) = n \times \frac{O(r_1)}{n} \times ... \times \frac{O(r_m)}{n}$$

Arboles de Decisión

Test Chi-cuadrado

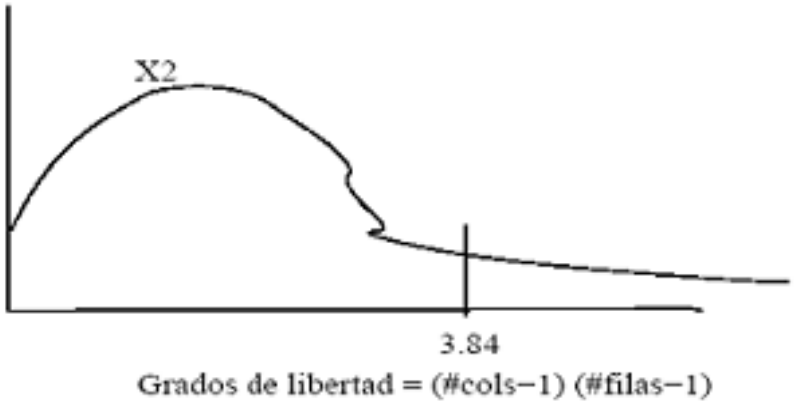
- Estimador: $\chi^2 = \sum_{r \in R} \frac{(O(r) - E(r))^2}{E(r)} \sim \chi^2_{(\#\text{col} - 1) \times (\#\text{filas} - 1)}$
- Un mayor indice de Chi-cuadrado indica una mayor dependencia entre la variable de la clase y la variable del split.
- Ejemplo: venta de té y café en un supermercado.
  - Tenemos dos variables Café (C) y Té (T).
  - Valores son compró o no-compró

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

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

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

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



El estimador Chi-Cuadrado es una desviación normalizada entre la tabla de contingencia real y la esperada

	Yes	No	Total
Sunny	2	3	5
Overcast	4	0	4
Rainy	3	2	5
Total	9	5	14

	Yes	No	Total
Sunny	3.21	1.79	5
Overcast	2.57	1.43	4
Rainy	3.21	1.79	5
Total	9	5	14

Arboles de Decisión

Test Chi-cuadrado

	Yes	No	Total
Sunny	0.46	0.83	
Overcast	0.79	1.43	
Rainy	0.01	0.03	
Total			3.55

- Grado de libertad=(3-1)*(2-1) =2
- Valor crítico a 95% de conf = 5.99
- Luego, no rechazamos la suposición de independencia
- El test de chi-cuadrado sólo se puede usar si no más del 10% de las celdas tienen menos de 5 de frecuencia observada.

Definición

Dado un espacio de modelos M, un modelo m en M es sobreajustado si existe otro modelo m' en M tal que:

- *m tiene menor error que m' en datos de entrenamiento*
- *m' tiene menor error que m en datos objetivo*

- Los árboles complejos (muy grandes) tienen la propiedad de que están sobre ajustados.
- La Poda es un mecanismo para obtener árboles con menor error de predicción
  - Pre-poda: Parar la construcción del árbol en algunas nodos
  - Post-poda: Construir un árbol complejo (posiblemente sobreajustado) y podarlo después.

Arboles de Decisión

Criterios de Poda

Pre-Poda

- Parar la construcción del árbol en algunos nodos en base a criterios:
  - No expandir si GiniSplit $<$ k
  - No expandir si el nodo tiene menos de k datos
  - No expandir si test de chi-cuadrado no rechaza independencia
- Al no expandir, etiquetar el nodo con la clase más frecuente en los datos asociados al nodo.

Post-Poda

- Poda Basada en reducción del error:
  - Podamos si el error de predicción del árbol podado disminuye
  - El error se estima usando un conjunto de datos de validación
- Poda Basada en Principio de Descripción Mínima
  - Se estima el costo de descripción del árbol y los datos, antes y después de eliminar el nodo.
  - El criterio es obtener el costo de descripción mínima

Arboles de Decisión

Algoritmo de Poda

- Operaciones de Poda
  - Reemplazo de un subárbol: Es la operación más común en métodos de poda y consiste en reemplazar un subárbol por una hoja que se etiqueta con la clase mayoritaria
  - Ascenso de un subárbol: Menos común, consiste en subir un subárbol a un nodo superior.
- Se recorre el árbol desde las hojas hacia arriba podando nodos.
- Para cada nodo que visitamos en el recorrido aplicamos un test para decidir si lo podamos o no.
- Primero estimación el error A si no se poda el nodo
  - Estimamos el error del subárbol bajo el nodo como la suma ponderada de errores estimados en nodos hijos
  - Esto se propaga de forma recursiva hasta las hojas, donde su error se estima usando inferencia estadística.
- Estimamos el error B si se poda el nodo
- Si B $<$ A, entonces se poda el nodo.

Arboles de Decisión

Estimación del Error de Predicción

- En general, el error de un nodo hoja se estima como el límite superior de un intervalo de confianza.
- La muestra de datos para calcular este intervalo se puede obtener de:
  - Los mismos datos de entrenamiento
  - Muestra disjunta de datos de entrenamiento: datos de poda
- Para un nodo hoja el error se estima como el límit superior de un intervalo de Wilson de un lado:

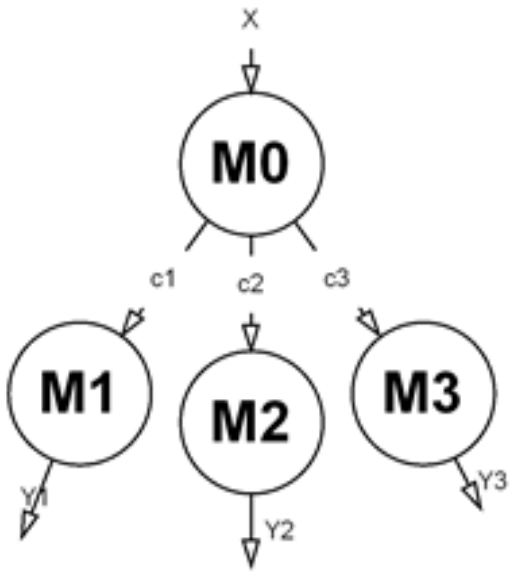
$$\Pr\left(\frac{p-\pi}{\sqrt{p(1-p)/n}} \leq z_{1-C}\right) = C$$

- Se obtiene el límite superior:

$$\pi = \frac{p + \frac{z^2}{2n} + z\sqrt{\frac{p}{n} - \frac{p^2}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}}$$

- Si C=75% de confianza, entonces $z = 0.69$

- Supongamos el caso que disponemos de los modelos M_1, M_2, M_3 para predecir una variable Y a partir de las variables $\vec{X}$.
- Para cada dato $d \in D$, tenemos un error e_i asociado al modelo M_i, i=1,2,3
- Para cada dato $d \in D$, le podemos crear la clase $c \in C = \{c_1, c_2, c_3\}$, donde $c = c_j$ implica que $e_j \leq e_i$, i=1,2,3
- Entonces, tomando los datos D, y reemplazando la variable Y por la clase C, tenemos un problema de clasificación.
- Luego, podemos crear un árbol de decisión M_0 que prediga la clase c a partir de las variables $\vec{X}$.
- Juntando este modelo M_0 con los modelos M_1, M_2, M_3, podemos crear un modelo M, tal que el error del modelo M es menor que el minimo de los errores de los modelos M_1, M_2, M_3



Además, se tiene que para cada dato $d \in D$, tenemos que el error e asociado al modelo M es $e = \min_i e_i$.

Luego, si creamos distintos modelos que se especializan en distintos subconjuntos de datos, entonces el modelo M tendrá un mayor poder predictivo, reduciendo su error, sin necesidad de sobreajuste.