



Entropía (Cont..)

- Entropía:

$$H(S) = - \sum_{c_i} p_i \log_2 p_i$$

- H(S) es la entropía del set S
 - c_i son las posibles clases
 - p_i = fracción de registros de S que tienen la clase C_i
- Ejemplo de entropía:
 - 3 clases (A,B,C)
 - A ocurre en la mitad de los ejemplos
 - B y C ocurren en un 1/4 de los ejemplos
 - Codificación óptima: A = 0, B = 10, C = 11
 - Entropía = número de bits promedio/registro = 1.5 bits

Karim Pichara B.

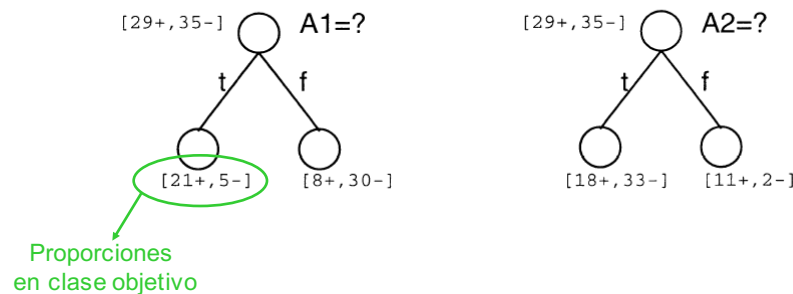
PUC Chile



Ganancia de la información

La ganancia de información es la reducción esperada en entropía al separar según cierto atributo, digamos A:

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$



Karim Pichara B.

PUC Chile



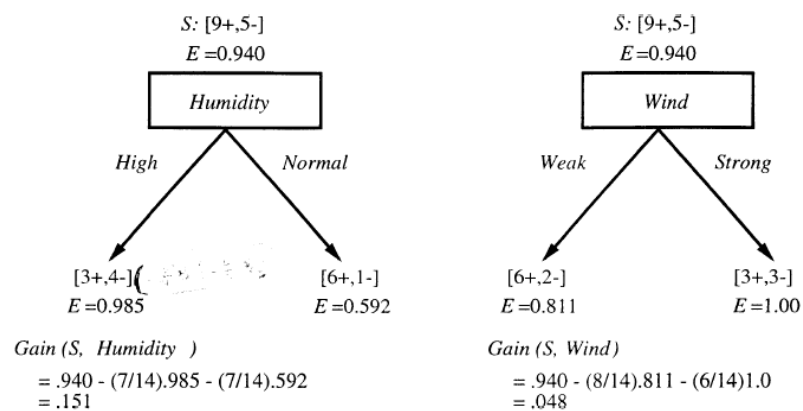
Ejemplo

Day	Outlook	Temperature	Humidity	Wind	PlayTenn
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

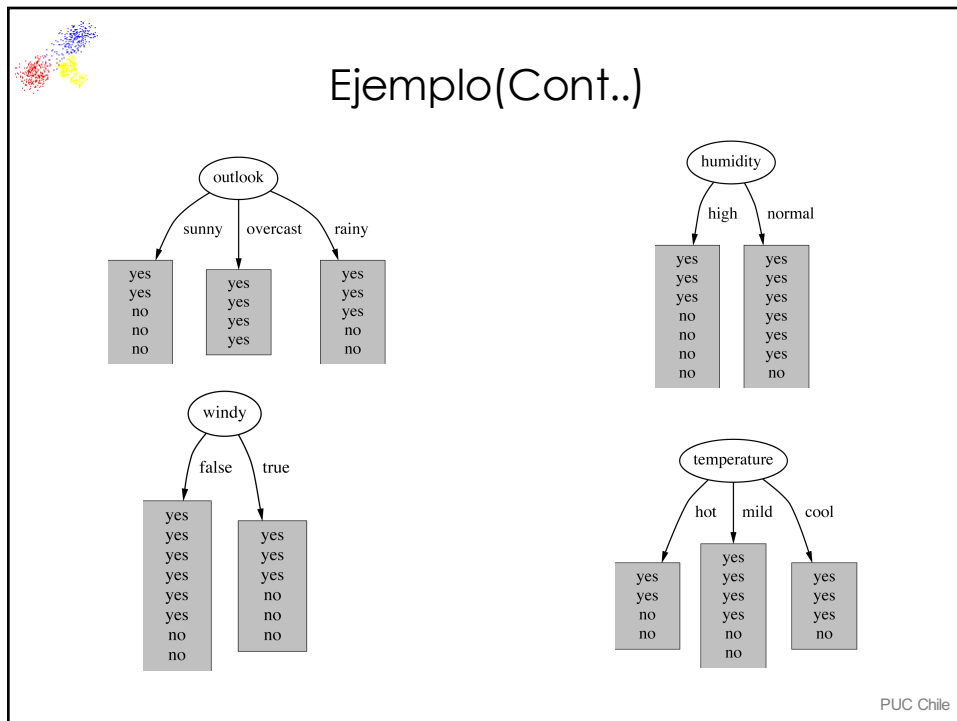
PUC Chile



Ejemplo



PUC Chile



outlook

- sunny
 - yes
 - yes
 - no
 - no
 - no
- overcast
 - yes
 - yes
 - yes
 - yes
- rainy
 - yes
 - yes
 - yes
 - no
 - no

humidity

- high
 - yes
 - yes
 - yes
 - no
 - no
 - no
 - no
- normal
 - yes
 - yes
 - yes
 - yes
 - yes
 - no
 - no

windy

- false
 - yes
 - yes
 - yes
 - yes
 - yes
 - yes
 - no
 - no
- true
 - yes
 - yes
 - yes
 - no
 - no
 - no

temperature

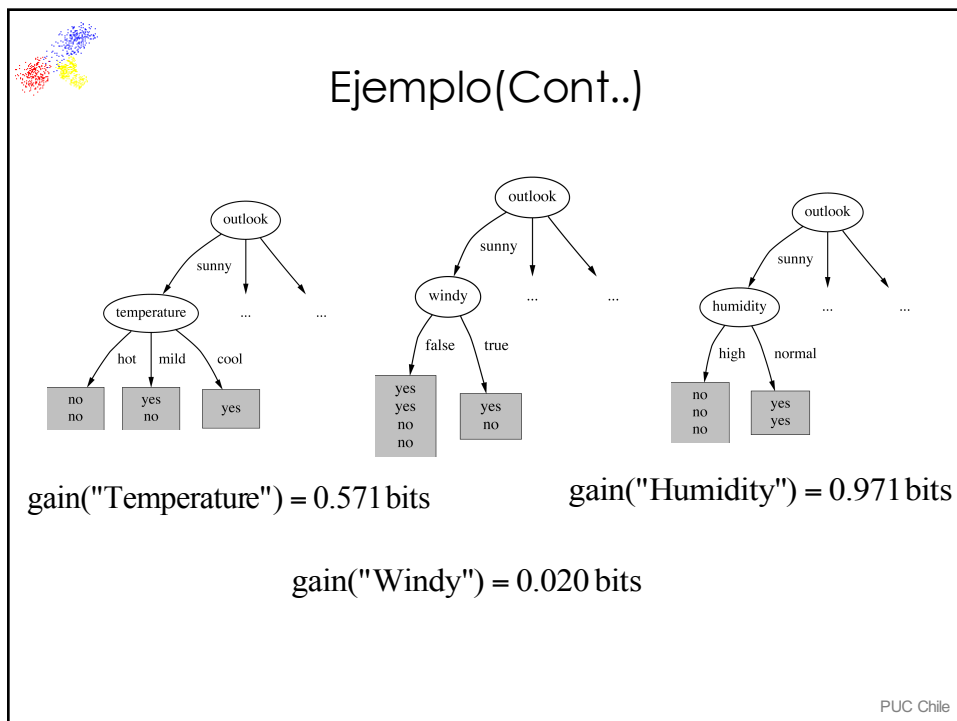
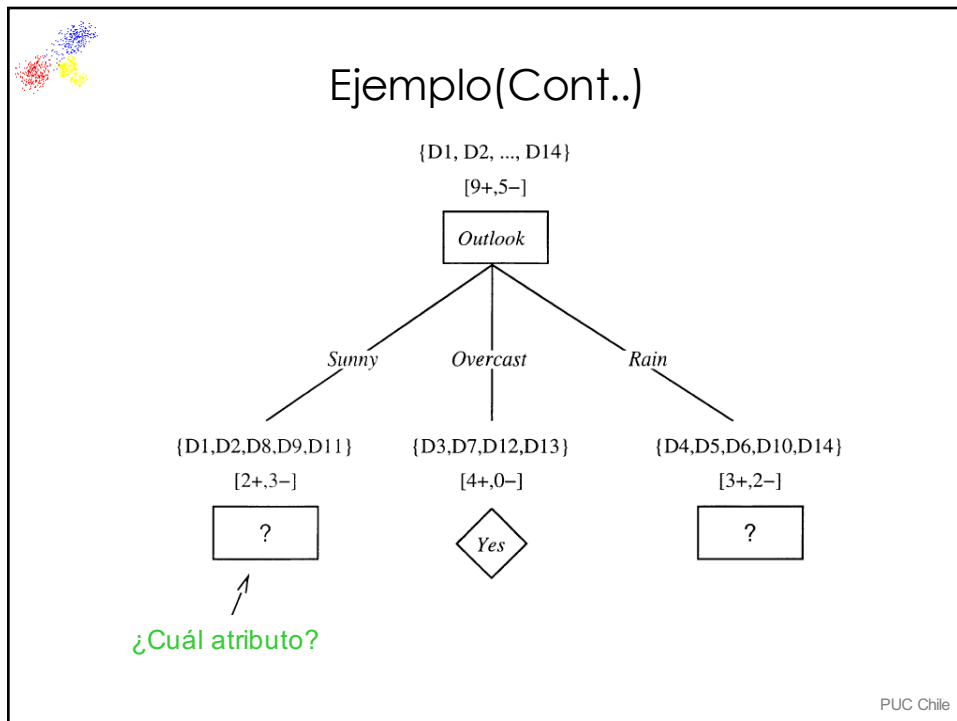
- hot
 - yes
 - yes
 - no
 - no
- mild
 - yes
 - yes
 - yes
 - yes
 - no
 - no
- cool
 - yes
 - yes
 - yes
 - no


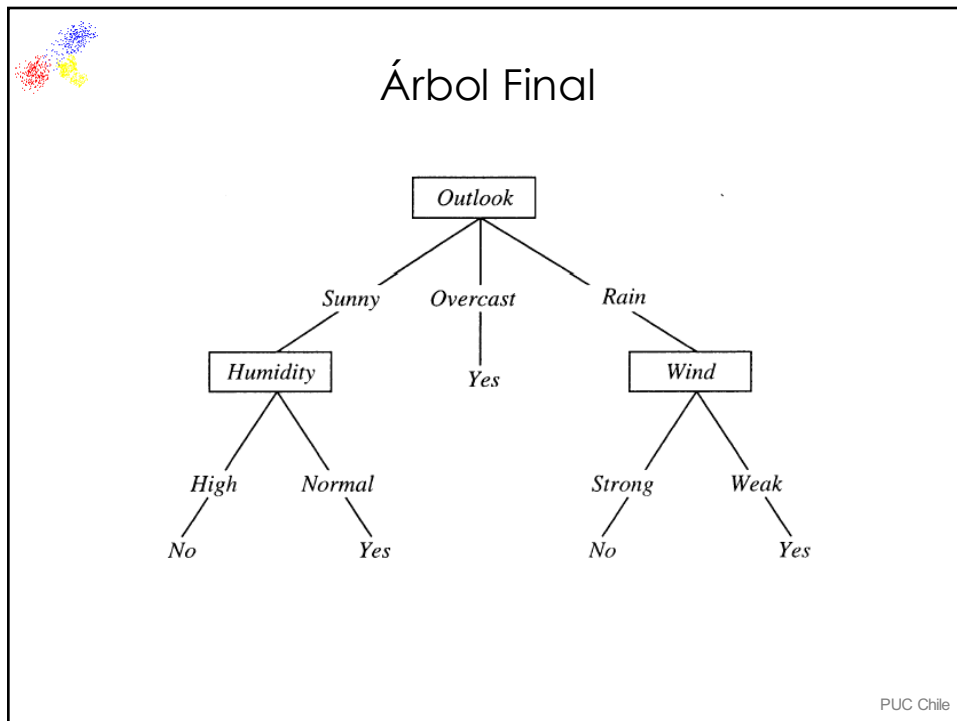
PUC Chile

Ejemplo(Cont..)

- “Outlook” = “Sunny”:
 $\text{info}([2,3]) = \text{entropy}(2/5, 3/5) = -2/5 \log(2/5) - 3/5 \log(3/5) = 0.971 \text{ bits}$
- “Outlook” = “Overcast”: $\text{info}([4,0]) = \text{entropy}(1,0) = -1 \log(1) - 0 \log(0) = 0 \text{ bits}$ *Consideramos: $0 * \log(0) = 0$*
- “Outlook” = “Rainy”:
 $\text{info}([3,2]) = \text{entropy}(3/5, 2/5) = -3/5 \log(3/5) - 2/5 \log(2/5) = 0.971 \text{ bits}$
- Expected information for attribute:
 $\text{info}([3,2], [4,0], [3,2]) = (5/14) \times 0.971 + (4/14) \times 0 + (5/14) \times 0.971 = 0.693 \text{ bits}$

PUC Chile

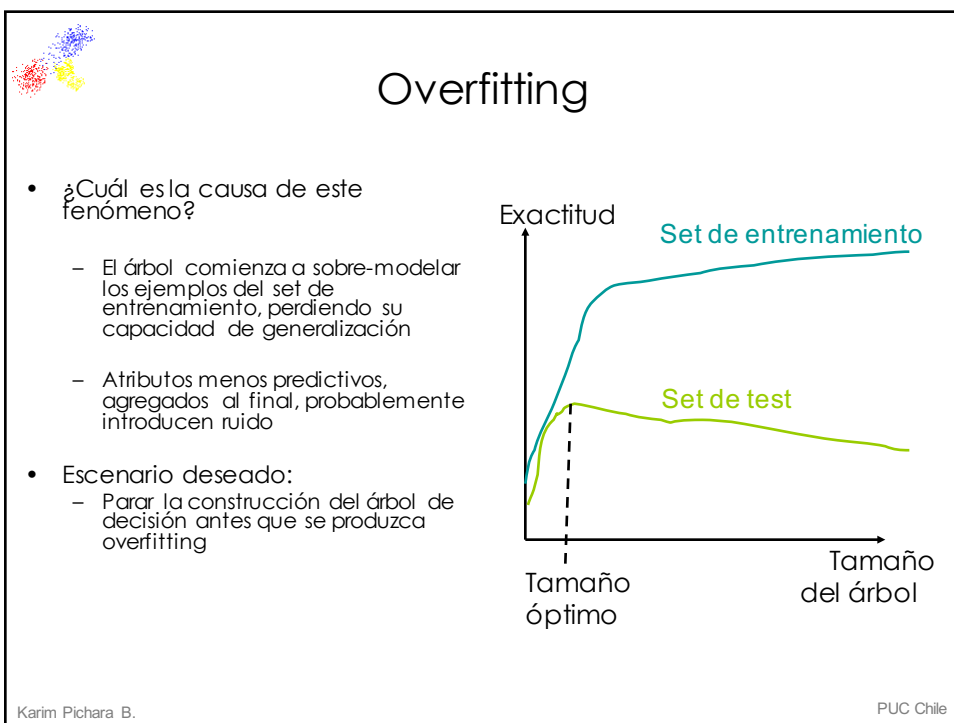
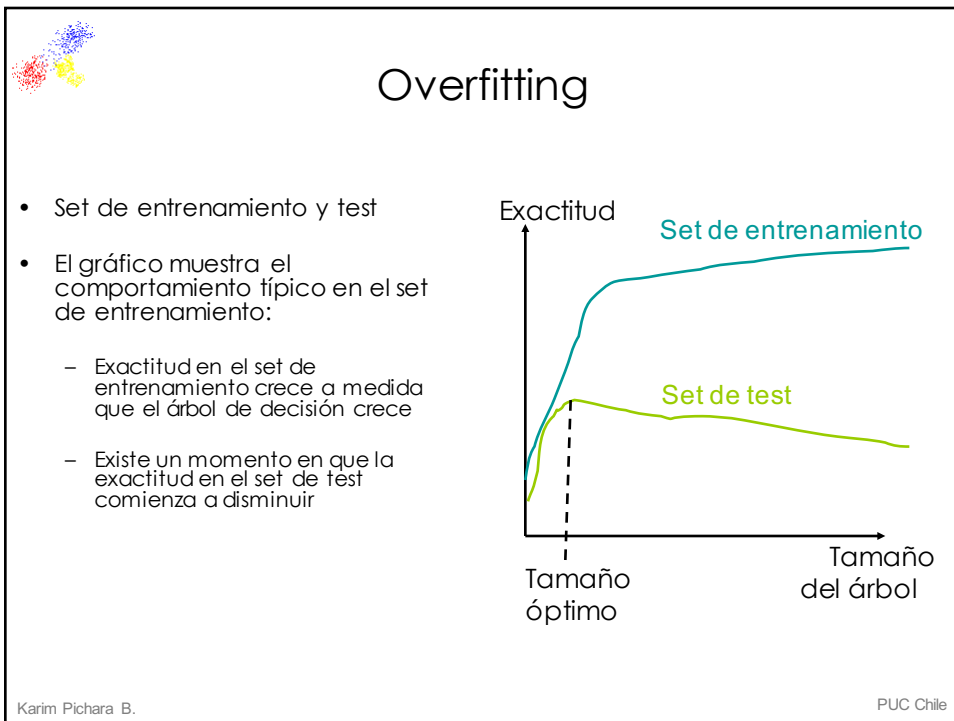




Consideraciones

- Una de las ventajas de los árboles de decisión es que entregan una mayor información sobre el proceso de toma de decisiones (no funciona como una caja negra)
- Una de las desventajas de este sistema es que no es capaz de considerar valores de un conjunto de atributos a la vez, sólo los va tomando de a uno

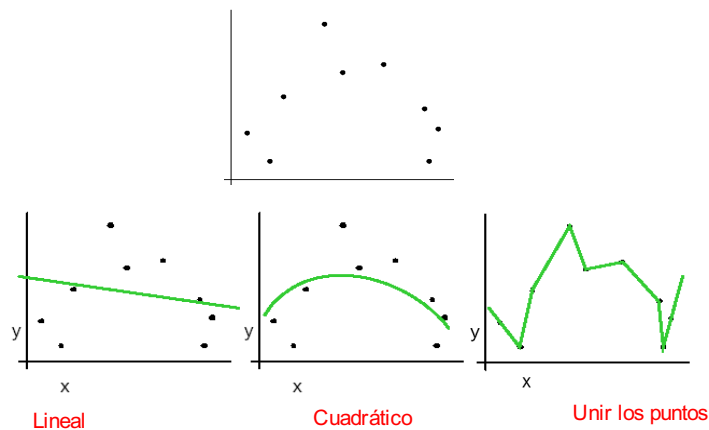
Karim Pichara B. PUC Chile





Overfitting

Overfitting es un problema con el cual deben lidiar la mayoría de los algoritmos de Data Mining



PUC Chile



Occam's Razor principle

Occam's razor is a logical principle attributed to the mediaeval philosopher William of Occam (or Ockham). The principle states that one should not make more assumptions than the minimum needed. It underlies all scientific modeling and theory building. It admonishes us to choose from a set of otherwise equivalent models of a given phenomenon the simplest one.





¿Cómo evitamos Overfitting?

- Parar la construcción del árbol cuando el número de registros restante no es estadísticamente significativo
- Construir el árbol sin considerar restricciones de tamaño y luego podar basándose en su rendimiento en el set de test
- Como seleccionar el mejor árbol
Métricas de evaluación del modelo (Ej: rendimiento en el set de test)

PUC Chile



Atributos con muchos valores

- Si un atributo tiene muchos valores, probablemente la métrica de ganancia de información lo seleccionará
– Ej. Día=Julio 7 2005
- Una forma de solucionar el problema es usar la razón de ganancia (GainRatio)

$$GainRatio(S, A) \equiv \frac{Gain(S, A)}{SplitInformation(S, A)}$$

$$SplitInformation(S, A) \equiv - \sum_{i=1}^c \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

PUC Chile



Ejemplo

rid	age	income	student	credit_rating	Class: buys_computer
1	<30	high	no	fair	no
2	<30	high	no	excellent	no
3	30-40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	30-40	low	yes	excellent	yes
8	<30	medium	no	fair	no
9	<30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<30	medium	yes	excellent	yes
12	30-40	medium	no	excellent	yes
13	30-40	high	yes	fair	yes
14	>40	medium	no	excellent	no

$$\text{SplitInfo}(S, \text{Income}) = -\frac{4}{14} \log_2 \frac{4}{14} - \frac{6}{14} \log_2 \frac{6}{14} - \frac{4}{14} \log_2 \frac{4}{14} = 1.5567$$

PUC Chile



Poda del Árbol

- Pre-podar: Este tipo de poda consiste en detener la expansión de un nodo en un momento dado de la construcción del árbol.
 - Una vez que se detiene la expansión se genera un nodo hoja con la clasificación más frecuente en el subconjunto de tuplas correspondiente.
- Post-Podar: Se genera el árbol completo y luego se buscan sub-ramas a podar, la poda se realiza de la misma forma que en el caso anterior

PUC Chile



Poda del Árbol (Cont..)

- Al momento de analizar cada nodo N compara la complejidad del sub-árbol desde el nodo N y la complejidad si se reemplaza el sub-árbol por una hoja.
- Se deben definir criterios de poda relativos a complejidad del árbol y reducción del error del set de test.

Karim Pichara B.

PUC Chile



Árboles de Decisión ¿cuándo?

- Tenemos datos de entrenamiento
- Función objetivo requiere clasificación con pocas clases
- Necesitamos generar reglas de decisión entendibles por personas (Un árbol de decisión es una disyunción de conjunciones)
- Atributos son discretos o discretizables sin gran pérdida de información

Karim Pichara B.

PUC Chile