# 2.3 APRENDIZAJE DE ÁRBOLES DE DECISIÓN

Francisco José Ribadas Pena, Santiago Fernández Lanza

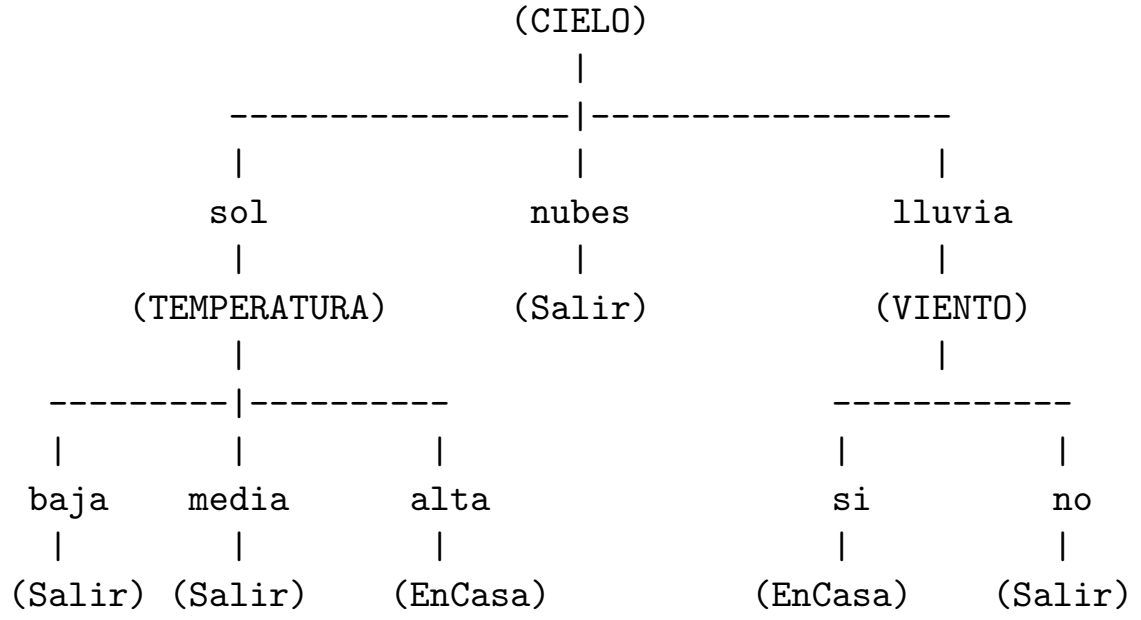
Modelos de Razonamiento y Aprendizaje
5º Informática
ribadas@uvigo.es, sflanza@uvigo.es

4 de febrero de 2013

# 2.3.1 INTRODUCCIÓN

**Árbol decisión:** Representación en forma de árbol de un clasificador de ejemplos en base a sus atributos.

- **hojas:** asignan clases
- **nodo internos:** nodos de decisión
  - especifican un test sobre un atributo
  - descendientes se corresponden con los diferentes valores posibles
    - arcos etiquetados con valores de atributo
- Ejemplo:




**Funcionamiento:** Toma como entrada un objeto o situación descrita por un conjunto de atributos y devuelve una clasificación.

- En principio: atributos y clases discretos (nominal), no continuos

**OBJETIVO**: Construir un árbol capaz de predecir la case de un ejemplo, dados

1. Conjunto de clases: $C = \{c_1, c_2, c_n\}$
2. Conjunto $E$, de ejemplos clasificado en forma de pares atributo-valor
   - uno de esos atributos es la clase

Potencia expresiva equivalente a lógica proposicional

- Se maneja sólo un objeto a la vez, no existe cuantificación
- Pares atributo-valor equivalen a lógica de proposiciones
  - notación más cómoda
  - compatibilidad BD relacionales

## Tipos de Aplicaciones

- Ejemplos pueden describirse mediante pares atributo valor
- Función objetivo discreta (conjunto de clases finito)
- Ejemplos: toma de decisiones (concesión créditos), predicción, ...

## Ventajas

- Técnica simple, pero relativamente potente
- Disponibilidad de algoritmos y problemas adecuados (info. en formato adecuado)
- Posibilidad de convertir árboles en reglas
  - Una regla por cada hoja
  - **Consecuente** = clase de la hoja
  - **Antecedente** = conjunción (AND) de todos los test efectuados en el camino desde la raíz hasta esa hoja

  Ejemplo:

```
IF (cielo=sol) AND (temp=baja) THEN (acion=salir)
IF (cielo=sol) AND (temp=media) THEN (accion=salir)
IF (cielo=sol) AND (temp=alta) THEN (accion=enCasa)
IF (cielo=nubes) THEN (accion=salir)
...
...
```

# 2.3.2 ALGORITMO DE CONSTRUCCIÓN

**OBJETIVO:** Encontrar el árbol más pequeño consistente con los ejemplos.

- Define un sesgo (preferencia) en la búsqueda de hipótesis
  - Sigue el principio conocido como *Navaja de Occam*:
    *"Ante distintas hipótesis consistentes con los datos, dar preferencia a la más simple."*
    - Árboles grandes $\Rightarrow$ simplemente "memorizan" ejemplos
    - Árboles pequeños $\Rightarrow$ mayor capacidad predicción (en principio)
- Es costoso encontrar el más pequeño $\Rightarrow$ uso de heurísticas (soluciones aproximadas)

**IDEA BASE:** Seguir aproximación descendente.

- Clasificar/Agrupar primero por atributos que diferencien mejor los ejemplos.
- Cada conjunto resultante es un nuevo problema (más simple)
- Aplicar esta misma idea recursivamente

### ALGORITMO

```
funcion ArbolDecicision(ejemplos, atributos): árbol
  SI todos los ejemplos en una Categoria
      DEVOLVER Hoja con esa Categoria
  SINO SI NO hay más atributos
      DEVOLVER Hoja con Categoría más común en los ejemplos
  SINO
      Seleccionar MEJOR ATRIBUTO (A)
      Etiquetar árbol con A
      PARA CADA Posible Valor de A (a_i)
         tomar ejemplos con ese valor (ejemplos_i)
         /* CREAR DESCENDIENTE  */
         llamada recurs. ArbolDecision(ejemplos_i, atributos - {A})
         enlazar con nodo A, etiquetando el enlace como "A=a_i"
      FIN PARA
  FIN SI
```

# 2.3.2.1 Elección Mejor Atributo

**PUNTO CLAVE:** Heurística de selección del MEJOR ATRIBUTO

- Valor máximo para atributos perfectos (discriminan perfectamente las clases)
- Valor mínimo para atributos no relevantes

**NOTA:** Aproximación básica similar a búsqueda por ascenso a colinas

- No garantiza el óptimo (mínimos locales)

**IDEA BASE (Algoritmo ID3, Quinlan 79):** Selección de atributos basada en la Teoría de la Información (Shanon y Weaver, 49)

- Conceptos de cantidad de información (entropía) y ganancia de información
- Selección del atributo que mejor separa los ejemplos (el que aporta mayor ganancia de información)

**Entropia:** (cantidad de información) Mide la *impureza* (desorden) de un conjunto de datos $D$

- Cantidad de información (en bits) que es necesaria para identificar una clase $C_i$ en $D$
- Inversamente proporcional a la frecuencia (probabilidad) de ocurrencia de dicha clase $C_i$.

Se mide en bits $\Rightarrow$ cálculo $= \boxed{-log_2(frecuencia(C_i))}$

- Para $k$ clases $(C_1, C_2, \ldots, C_k)$, la entropia del conjunto $D$ es la suma ponderada de las entropias de cada clase

$$\boxed{entropia(D) = \sum_{i=1}^{k} frecuencia(C_i) \times (-log_2(frecuencia(C_i)))}$$

**Ganancia de información:** Reducción de la entropía después de clasificar $D$ en base a un atributo $A$

- Diferencia entre la entropía de los subconjutos resultantes de clasificar $D$ de acuerdo los valores de un atributo $A$ y la entropía de $D$ sin clasificar.
- Siendo $a_1, a_2, ..., a_k$ los posibles valores de $A$ y $D_i$ los subconjuntos de $D$ asociado a cada valor $a_i$:

$$ganancia(D, A) = entropia(D) - \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times entropia(D_i)$$

Para elegir el mejor atributo en cada paso del algoritmo anterior:

- Para cada atributo no utilizado $\rightarrow$ calcular su ganancia sobre el conjunto de ejemplos actual
- Quedarse con el atributo que proporcione mayor ganancia

**EJEMPLO:**

| Cielo | Temp | Viento | Acción |
|---|---|---|---|
| Sol | Media | Si | Salir |
| Sol | Alta | Si | En Casa |
| Sol | Alta | No | En Casa |
| Sol | Media | No | Salir |
| Sol | Baja | No | Salir |
| Nubes | Media | Si | Salir |
| Nubes | Alta | No | Salir |
| Nubes | Baja | Si | Salir |
| Nubes | Alta | No | Salir |
| Lluvia | Media | Si | En Casa |
| Lluvia | Baja | Si | En Casa |
| Lluvia | Media | No | Salir |
| Lluvia | Baja | No | Salir |
| Lluvia | Media | No | Salir |

Entropia de los datos (2 clases: Salir/EnCasa)

$$entr(D) = \frac{4}{14}\left(-\log_2 \frac{4}{14}\right) + \frac{10}{14}\left(-\log_2 \frac{10}{14}\right) = 0{,}86 \text{ bits}$$

1. Clasificación por **Cielo** (tres valores: Sol, Nubes, Lluvia)

$entr(D_{\text{Sol}}) = \frac{3}{5}\left(-\log_2 \frac{3}{5}\right) + \frac{2}{5}\left(-\log_2 \frac{2}{5}\right) = 0{,}97$
$entr(D_{\text{Nubes}}) = \frac{4}{4}\left(-\log_2 \frac{4}{4}\right) + \frac{0}{4}\left(-\log_2 \frac{0}{4}\right) = 0$ (ya clasificado)
$entr(D_{\text{Lluvia}}) = \frac{2}{5}\left(-\log_2 \frac{2}{5}\right) + \frac{3}{5}\left(-\log_2 \frac{3}{5}\right) = 0{,}97$

$$gan(D, Cielo) = entr(D) - \left(\frac{5}{14}entr(D_{\text{Sol}}) + \frac{4}{14}entr(D_{\text{Nubes}}) + \frac{5}{14}entr(D_{\text{Lluvia}})\right) = \\ = 0{,}86 - 0{,}69 = 0{,}17 \text{ bits}$$

2. Clasificación por **Temperatura** (tres valores: Baja, Media, Alta)

$entr(D_{\text{Baja}}) = \frac{3}{4}\left(-\log_2 \frac{3}{4}\right) + \frac{1}{4}\left(-\log_2 \frac{1}{4}\right) = 0{,}81$
$entr(D_{\text{Media}}) = \frac{5}{6}\left(-\log_2 \frac{5}{6}\right) + \frac{1}{6}\left(-\log_2 \frac{1}{6}\right) = 0{,}65$
$entr(D_{\text{Alta}}) = \frac{2}{4}\left(-\log_2 \frac{2}{4}\right) + \frac{2}{4}\left(-\log_2 \frac{2}{4}\right) = 1$

$$gan(D, Temp) = entr(D) - \left(\frac{4}{14}entr(D_{\text{Baja}}) + \frac{6}{14}entr(D_{\text{Media}}) + \frac{4}{14}entr(D_{\text{Alta}})\right) = \\ = 0{,}86 - 0{,}79 = 0{,}07 \text{ bits}$$

3. Clasificación por **Viento** (dos valores: Si, No)

$$entr(D_{\text{Si}}) = \frac{3}{6}\left(-\log_2 \frac{3}{6}\right) + \frac{3}{6}\left(-\log_2 \frac{3}{6}\right) = 1$$
$$entr(D_{\text{No}}) = \frac{7}{8}\left(-\log_2 \frac{7}{8}\right) + \frac{1}{8}\left(-\log_2 \frac{1}{8}\right) = 0{,}54$$

$$gan(D, Viento) = entropia(D) - \left(\frac{6}{14} entr(D_{\text{Si}}) + \frac{8}{14} entr(D_{\text{No}})\right) = 0{,}86 - 0{,}74 = 0{,}12 \text{ bits}$$

Atributo Mayor Ganancia: **Cielo** ⇒ Clasificar por Cielo ⇒ Resultan tres subproblemas

*Cielo = Sol*

| Temp | Viento | Acción |
|---|---|---|
| Media | Si | Salir |
| Alta | Si | EnCasa |
| Alta | No | EnCasa |
| Media | No | Salir |
| Baja | No | Salir |

Siguiente clasificación: **Temp**

*Cielo = Nubes*

| Temp | Viento | Acción |
|---|---|---|
| Media | Si | Salir |
| Alta | No | Salir |
| Baja | Si | Salir |
| Alta | No | Salir |

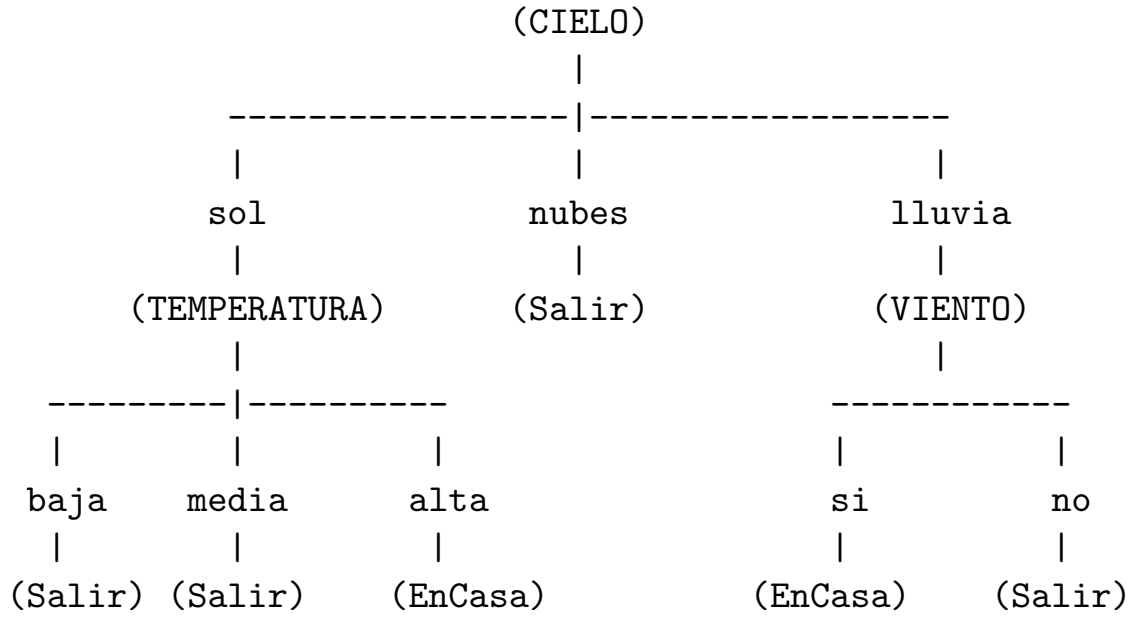
Ya clasificado (etiqueta = **Salir**)

*Cielo = Lluvia*

| Temp | Viento | Acción |
|---|---|---|
| Media | Si | EnCasa |
| Baja | Si | EnCasa |
| Media | No | Salir |
| Baja | No | Salir |
| Media | No | Salir |

Siguiente clasificación: **Viento**

Árbol final



# 2.3.3 PROBLEMAS

## (a) Sobreadaptación y ruido

**Problema:** Un árbol que clasifique los datos de entrenamiento perfectamente, no asegura que vaya a proporcionar la mejor generalización.

**DEF: Ruido** Existe *ruido* cuando hay ejemplos erróneos o contradictorios en los datos de entrenamiento.

- Si hay ruido en los datos de entrenamiento.
  El árbol se ajustará para cubrir esos casos $\Rightarrow$ árbol crece

**Problema:** Las decisiones realizadas en los niveles inferiores pueden estar soportadas por un conjunto de datos demasiado pequeño, no representativo de los datos reales.

**DEF: Sobreadaptación**

> Existe *sobreadaptación* de una hipótesis $h$ ante un conjunto de datos de entrenamiento, cuando existe otra hipótesis $h'$ cuyo error en el conjunto de entrenamiento es mayor que el de $h$, pero sucede que el error de $h$ sobre el conjunto de prueba es mayor que el de $h'$.

Puede darse en cualquiera de los métodos de aprendizaje.

- $h$ "cubre" muy bien los ejemplos con los que fue entrenada, pero no generaliza para soportar otros ejemplos distintos.
- Normalmente la presencia de "ruido" (ejemplos erroneos o contradictorios) ocasiona *sobreadaptación*
  - En árboles de decisión: en los niveles más bajos se añadirán nodos única y exclusivamente para "cubrir" esos ejemplos erróneos.

**Solución:** Eliminar algunos nodos de niveles inferiores (podar).

- **pre-poda:** (prepruning) detener el crecimiento de ciertos nodos durante la construcción del árbol
- **post-poda:** (postpruning) eliminar determinados nodos después de construida la totalidad del árbol

Elección de nodos a podar:

- Fijar cotas de profundidad máxima (llegado a una profundidad límite no dividir más nodos)
- Estudio estadístico: Podar cuando no hay un número mínimo de ejemplos en una rama que permitan hacer decisiones suficientemente fundamentadas:
  - Cortar cuando al añadir nuevos tests no mejore sustancialmente la clasificación global de los ejemplos de entrenamiento

# (b) Necesidad de Otras Heurísticas

**Problema:** Selección de atributos basada en la ganancia de información favorece atributos con muchos valores diferentes.

- Dividen datos en conjuntos muy pequeños
- Conjuntos pequeños $\Rightarrow$ datos más uniformes (baja entropía)
- Ejemplos: nombres, IDs, fechas, etc,...

Situar atributos con muchos valores en los nodos superiores hacen el árbol menos general

- Clasificación basada en valores muy específicos de un sólo atributo
- Degenera en memorización de ejemplos si hay un sólo ejemplo para cada valor del atributo (árboles de un sólo nivel)
- Menor poder predictivo

**Solución:** Ganancia ponderada. (Algoritmo C4.5, Quinlan)

- Normalizar la ganancia dividiendo por la entropía repecto a los valores del atributo, no respecto a las clases
- Para un atributo $A$ con $n$ valores, $a_1, a_2, \ldots, a_n$.
  La entropía de $D$ respecto a $A$ es:

$$entropia_A(D) = \sum_{i=1}^{n} frecuencia(a_i) \times (-log_2(frecuencia(a_i)))$$

  - Mide la información que ofrece la división de $D$ respecto a los valores de un atributo $A$
  - Atributos con muchos valores en pocos ejemplos $\Rightarrow$ cantidad de información alta
- La ganancia ponderada se calcula como:

$$ganancia_{pond}(D, A) = \frac{ganancia(D,A)}{entropia_A(D)}$$

- Penaliza ganancia de atributos con muchos valores
- Asegura que la división sea realmente informativa (útil)
- En la práctica se combinan las dos ganancias
  - Uso de ganancia poderada con atributos que superen un límite de valores posibles

# 2.3.4 EXTENSIONES

**(a) Atributos continuos** En principio áboles de decisión sólo manejan atributos nominales.

Para manejar valores numéricos continuos $\Rightarrow$ *discretizar* agrupando valores en rangos

Ej.:

```
temperatura: 18°, 19.5°, 28.5°, 32°,...
rangos: 0-10, 10-20, 20-30, 30-40
temperatura: baja, media, alta, muy alta
```

Selección de valores límite $\Rightarrow$ usar *ganancia de información*

- Seleccionar distintos límites, y elegir conjunto de límites que ofrezcan mayor ganacia (agrupan mejor los ejemplos)
- Elección de límites:
  - ordenar los valores del atributo continuo $A$ presentes en los ejemplos .
  - elegir como límites el punto medio entre dos valores cuya clase cambie

## (b) Valores desconocidos

En la prática es posible que no dispongamos el valor de algún atributo

- Ejemplo: imposible prueba forense si no se dispone del cadáver
- Puede suceder en el conj. de entrenamiento o al clasificar los ejemplos reales

Aproximación usual: "replicar" el ejemplo entre todos los posibles valores del atributo, ponderando cada "réplica" en base a la frecuencia de aparición de cada valor bajo un nodo dado

- En entrenamiento: (ejemplo)

```
En nodo raiz: frecuencia sol=5/14=0.35, nubes=4/14=0.29, lluvia=5/14=0.35

Un ejemplo <cielo=?, temp=alta, viento=si, accion=salir>
Se verá como tres ejemplos ‘‘incompletos’’ :
Para el cálculo de ganacias
   cuenta 0.35 veces como <cielo=sol, temp=alta, viento=si, accion=salir>
   cuenta  0.29 veces como <cielo=nubes, temp=alta, viento=si, accion=salir>
   cuenta 0.35 veces como <cielo=lluvia, temp=alta, viento=si, accion=salir>
```

- En clasificación: se "divide" el ejemplo y se clasifica cada "réplica", se ponderan las posibles clasificación en base a las frecuencias
  - "gana" la clasificación con más peso

# (c) Atributos con coste

(Relacionado con lo anterior.)

En ciertos dominios puede ser más costoso obtener el valor de unos atributos que el de otros, aun cuando los más costosos sean más informativos.

- Ejemplo: preferir hacer primero análisis de sangre, antes que resonancia magnética o biopsia

**Idea:** ponderar atributos de forma que se de preferencia a los test sobre atributos menos costosos

- Objetivo: minimizar el coste medio necesario para clasificar un ejemplo
- Ajustar medida de *ganacia* para incluir el coste
- Fórmula usual:

$$ganancia_{con\_costes}(D, A) = \frac{ganacia^2(D,A)}{coste(A)}$$