Sistemas Inteligentes

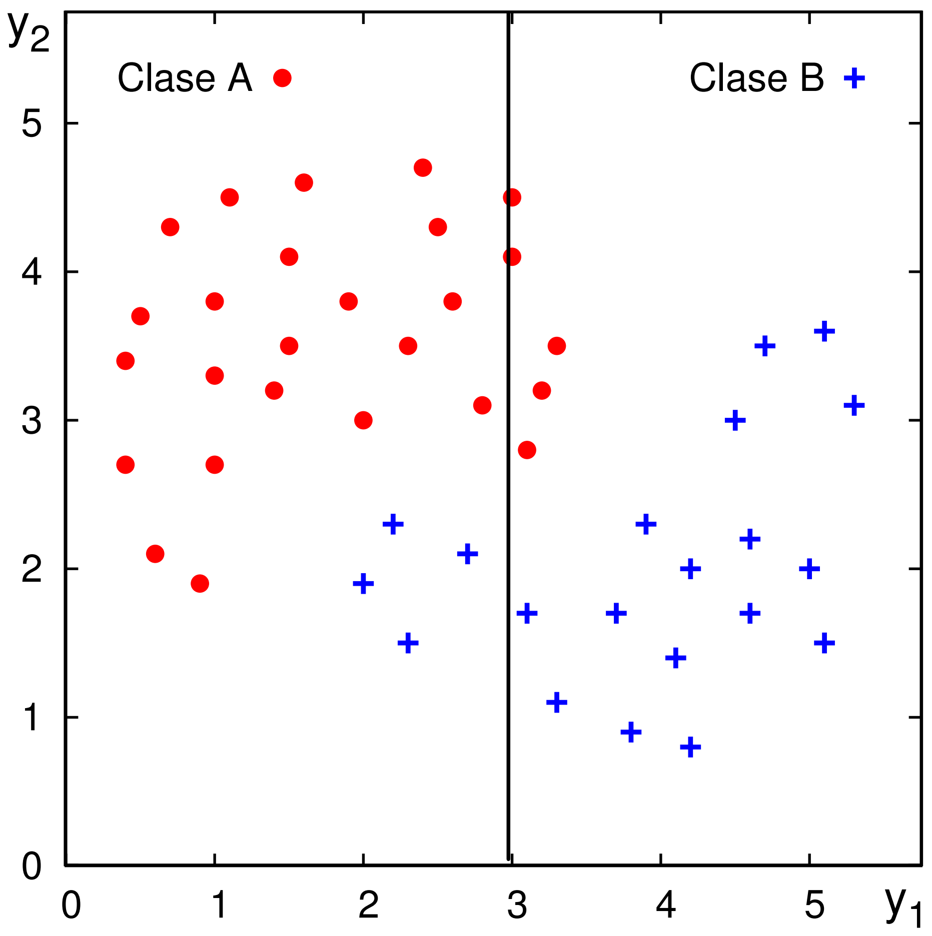
Escuela Técnica Superior de Informática

Universitat Politècnica de València

Tema B2T3
Árboles de Clasificación

Índice

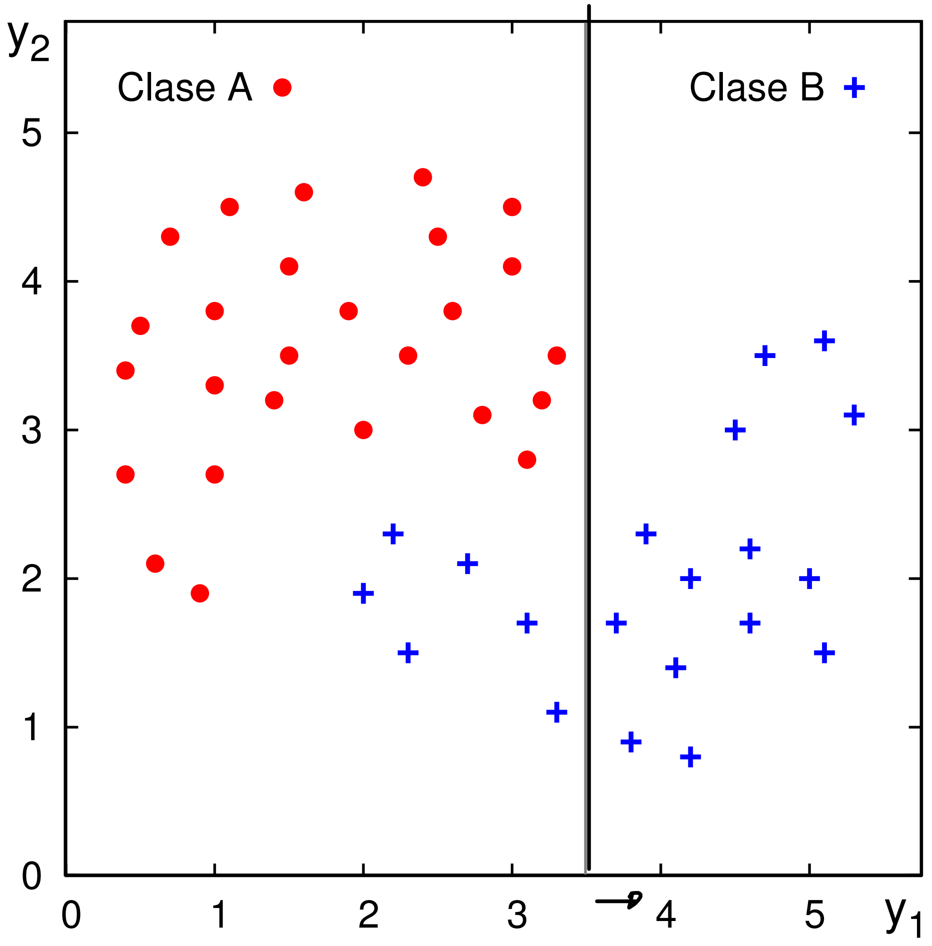
Ejemplo

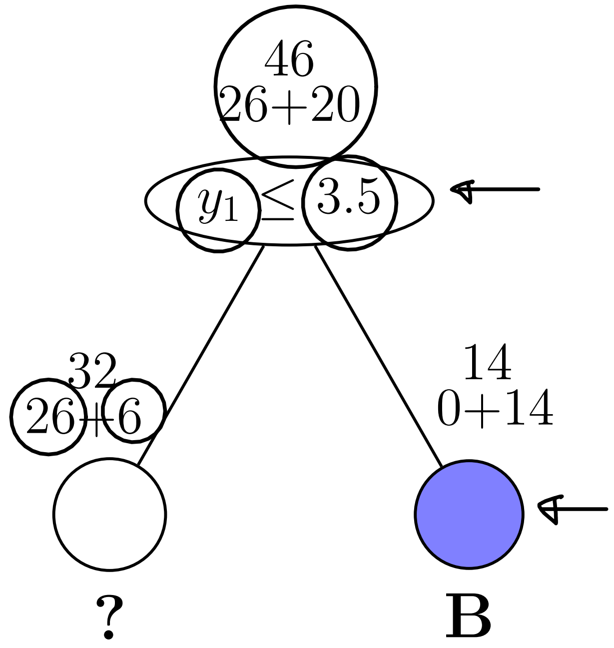


Tarea simple ilustrativa:

- representación en dos dimensiones ($E = \mathbb{R}^2$)
- clasificación en 2 clases *no* separables linealmente
- 46 datos (vectores)
  - 26 de clase A
  - 20 de clase B

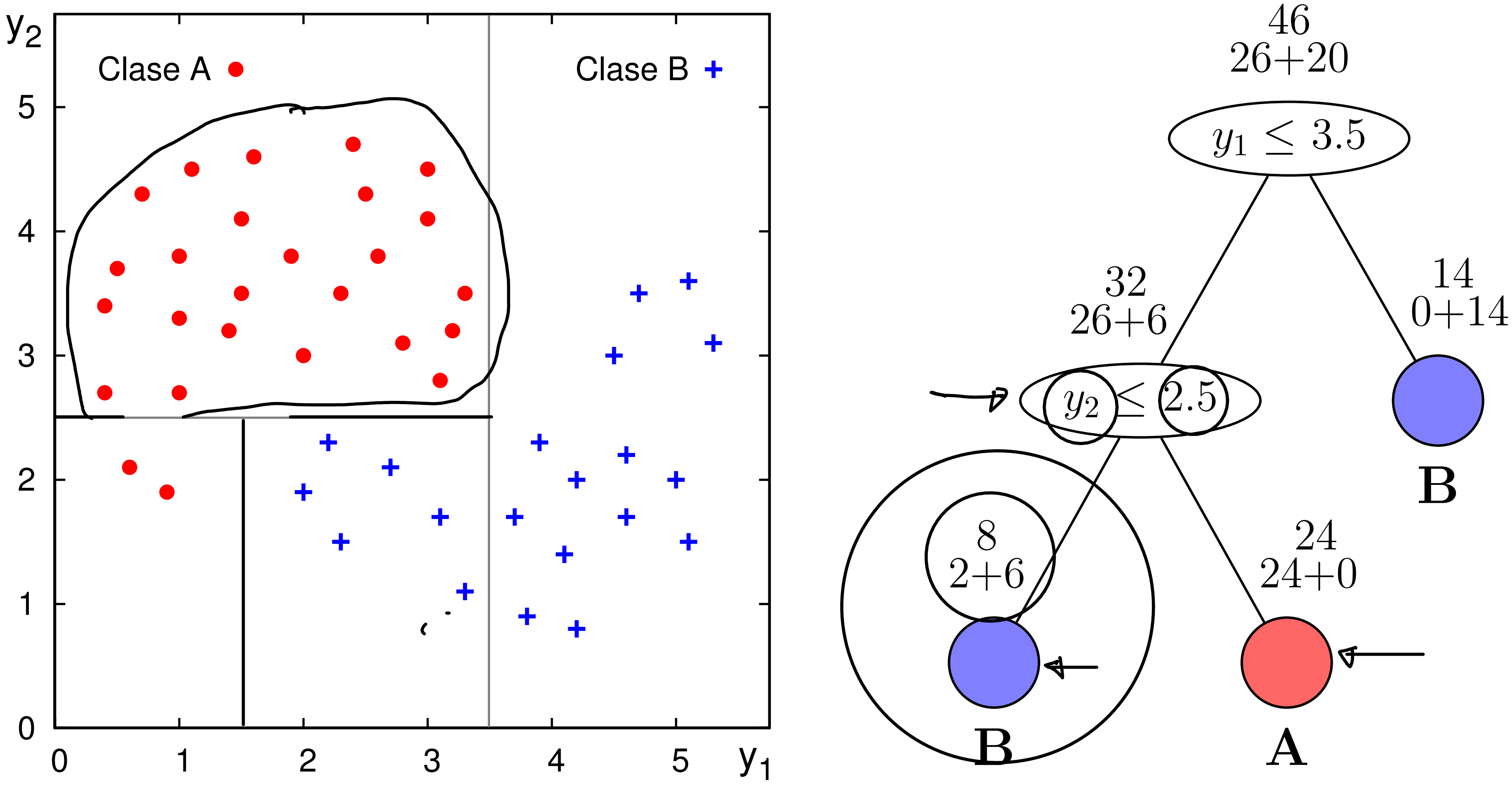
Ejemplo: Primera partición





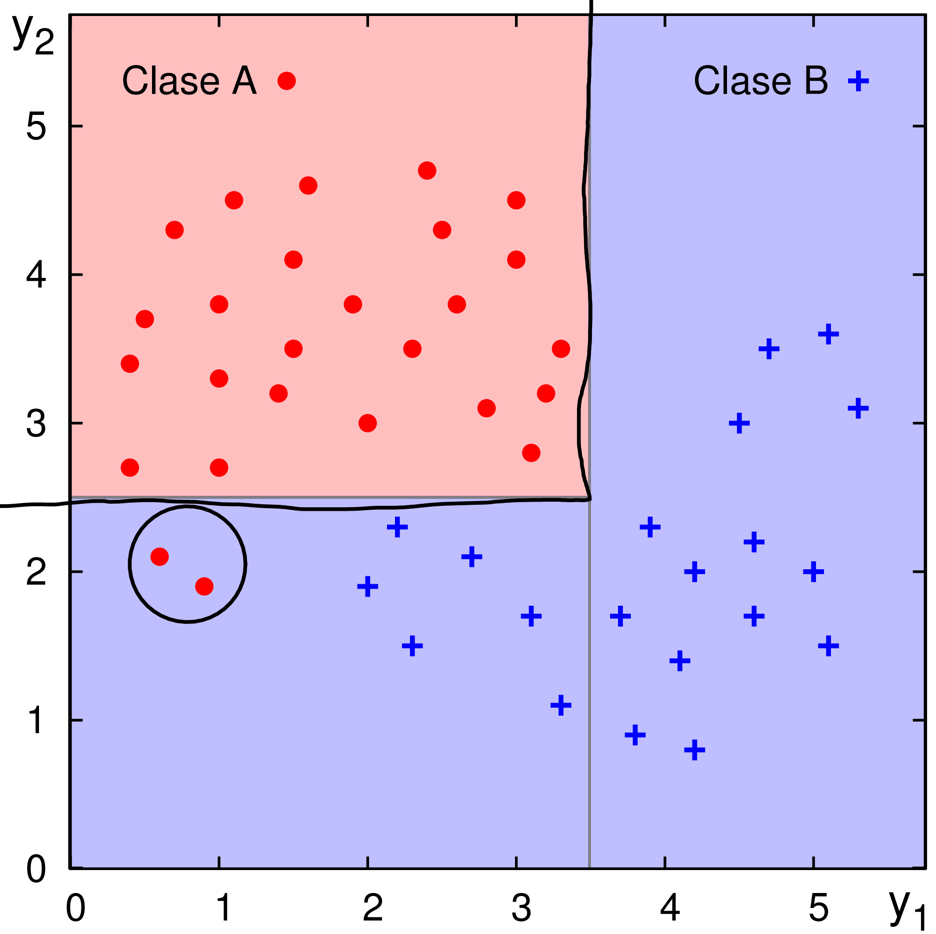
- El nodo raíz evalúa todos los datos con ¿$y_1 \leq 3.5$?
- El nodo derecho es *puro* porque todos los datos de ese nodo son de clase B.

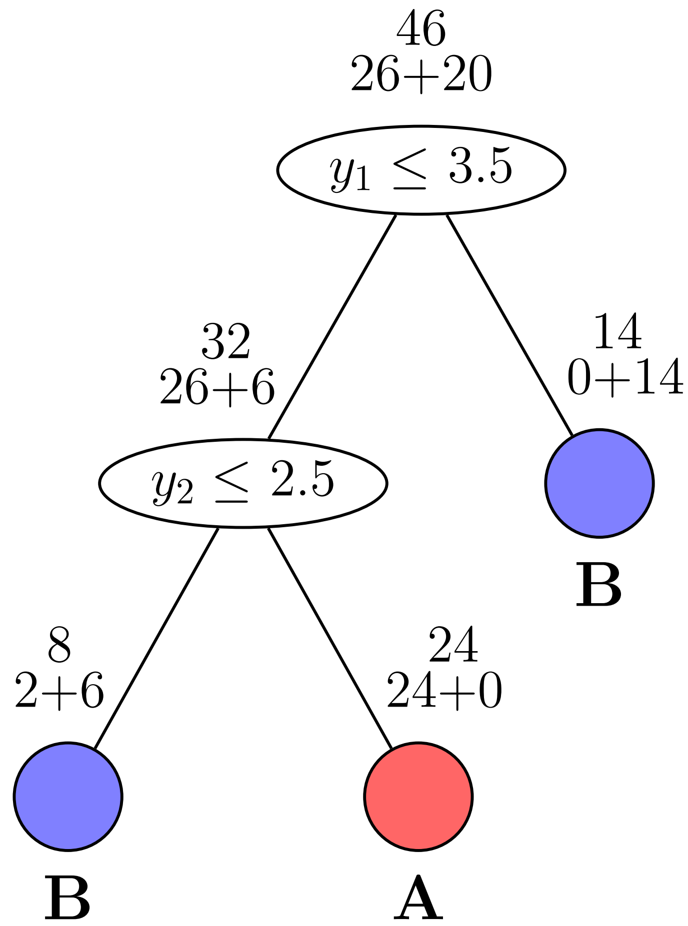
Ejemplo: segunda partición y fronteras de decisión



- Los datos del nodo izquierdo se particiona con ¿$y_2 \leq 2.5$?
- El nodo-hijo derecho es *puro* y se etiqueta como clase A
- El izquierdo no se particiona más y se etiqueta como B (clase mayoritaria)

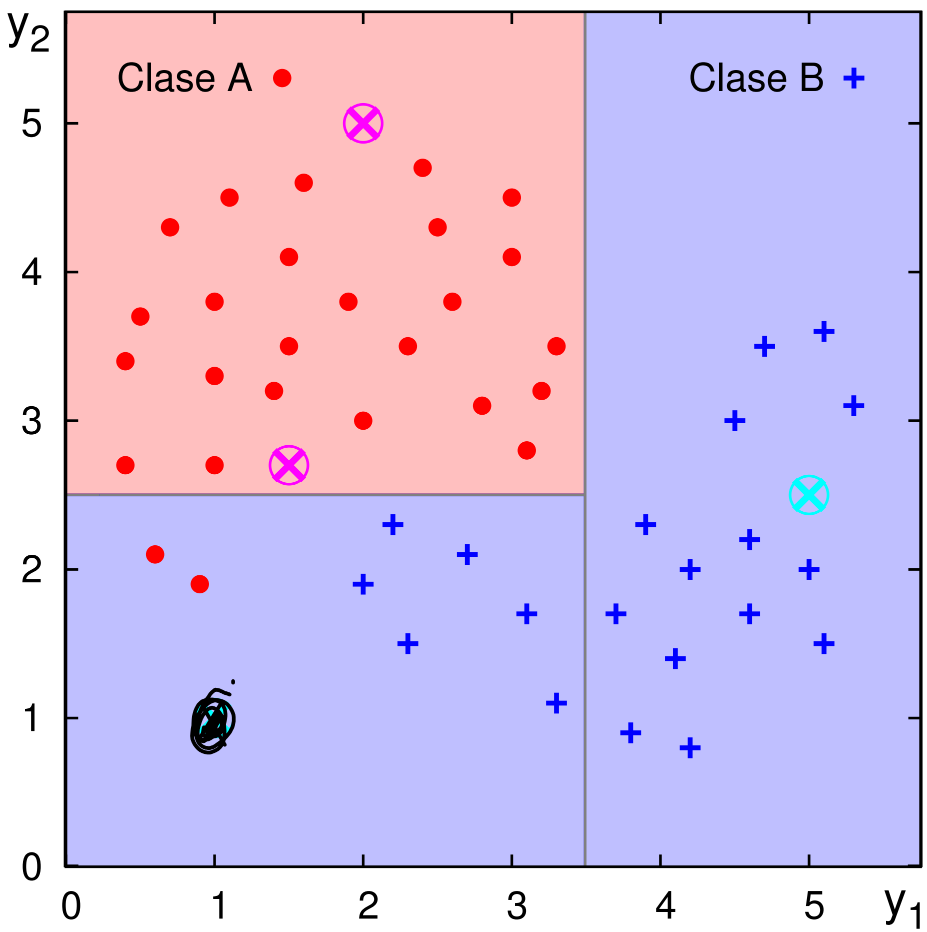
Ejemplo: regiones de decisión

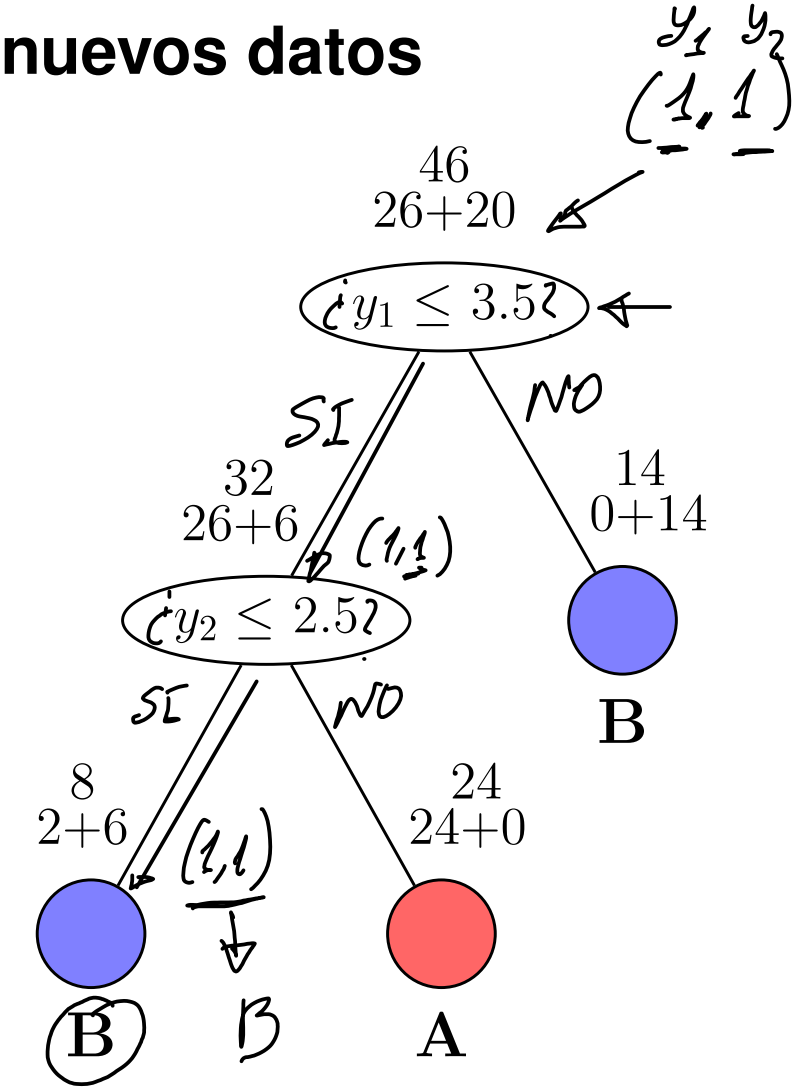




- Fronteras de decisión paralelas a los ejes
- Regiones de decisión rectangulares.
- La probabilidad de error estimada por resustitución es $2/46 = 0.0435 \rightarrow 4.35\,\%$

Ejemplo: clasificación de nuevos datos





El árbol de decisión obtenido permite clasificar nuevos datos:

$\boxed{(1.0, 1.0)^t}$: $y_1 \leq 3.5, y_2 \leq 2.5 \rightarrow$ clase B $\qquad$ $(1.5, 2.7)^t$: $y_1 \leq 3.5, y_2 > 2.5 \rightarrow$ clase A

$(5.0, 2.5)^t$: $y_1 > 3.5 \rightarrow$ clase B $\qquad$ $(2.0, 5.0)^t$: $y_1 \leq 3.5, y_2 > 2.5 \rightarrow$ clase A

... $\qquad$ ...

Árboles de clasificación (ADC)

- Estructura resultante de la *partición recursiva* del *espacio de representación* a partir de una *datos de aprendizaje*.

- Cada nodo interno evalúa con una pregunta un atributo concreto

- Cada nodo hoja contiene una etiqueta de clase (clasificación)

- Los datos de test se clasifican mediante una serie de preguntas sobre los valores de sus atributos, empezado por el nodo raiz y siguiendo el camino determinado por las respuestas a las preguntas de los nodos internos, hasta llegar a un nodo hoja que clasifica la muestra en una clase

- Veremos CART[1]caracterizado por una partición binaria de los nodos basada en criterios estadísticos sólidos.

[1]Classification And Regression Trees

Notación

- Espacio de representación: $E \equiv \mathbb{R}^D$; $\boldsymbol{y} = (y_1, y_2. \ldots, y_D)^t \in E$

- Muestra de aprendizaje: N vectores, con su correcta clasificación:
$(\boldsymbol{y}_1, c_1), \ldots, (\boldsymbol{y}_N, c_N), \quad \boldsymbol{y}_i \in E,\ c_i \in \mathbb{C} = \{1, 2, \ldots, C\},\ 1 \leq i \leq N$

- Un árbol se denota por T (*"Tree"*), un nodo por t, sus hijos izquierdo y derecho por t_L, t_R, respectivamente y el conjunto de nodos-hoja o terminales por $\tilde{T}$

- Una partición binaria (*"split"*) se denota por s y el conjunto de particiones admisibles por S

Estimación de probabilidades asociadas a los nodos de un ADC

Sean: $\boxed{N}$, $\boxed{N_c}$, $\boxed{N(t)}$, $\boxed{N_c(t)}$, respectívamente, el número total de datos de la muestra de aprendizaje, el número de estos datos de la clase c, el número de los que están representados en el nodo t, y el número de estos últimos que son de la clase c.

Probabilidad a priori de la clase c : $\underline{\hat{P}(c)} = \dfrac{N_c}{N}$

Probabilidad a posteriori de clase en el nodo t : $\underline{\hat{P}(c \mid t)} = \dfrac{N_c(t)}{N(t)}$

Probabilidad de un nodo *terminal*, $t \in \tilde{T}$: $\hat{P}(t) = \dfrac{N(t)}{N}$

Probabilidad de decisión por el hijo izquierdo de t : $\hat{P}_L(t) = \dfrac{N(t_L)}{N(t)}$

Probabilidad de decisión por el hijo derecho de t : $\hat{P}_R(t) = \dfrac{N(t_R)}{N(t)}$

10

N(t)

N(t_L) N(t_R)

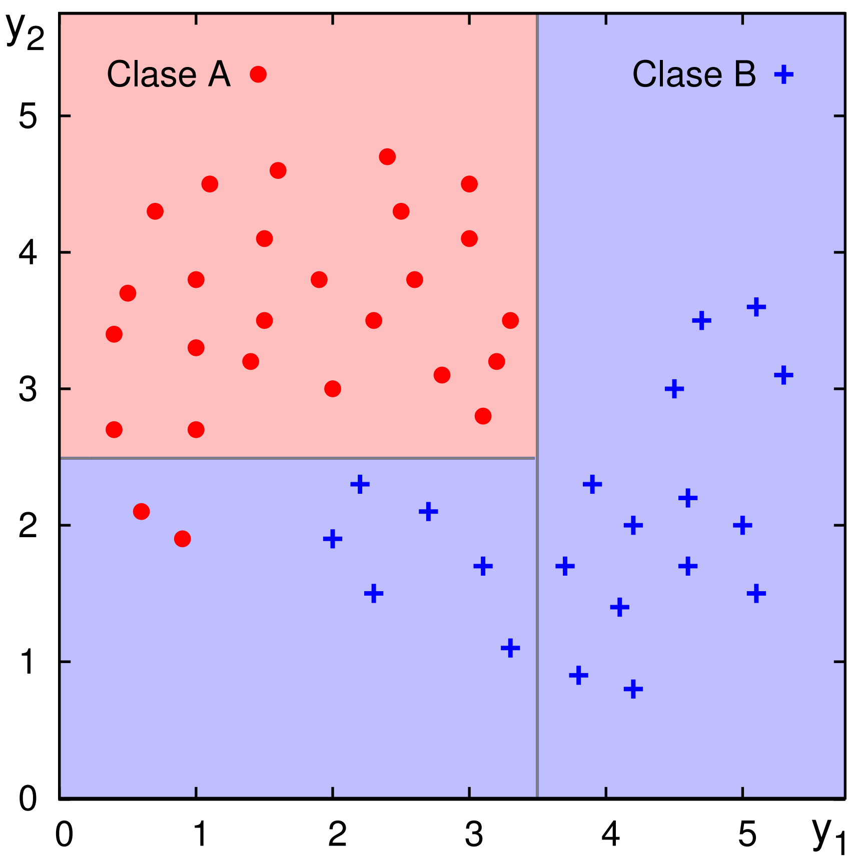
3 7

$\frac{3}{10} + \frac{7}{10} = 1$

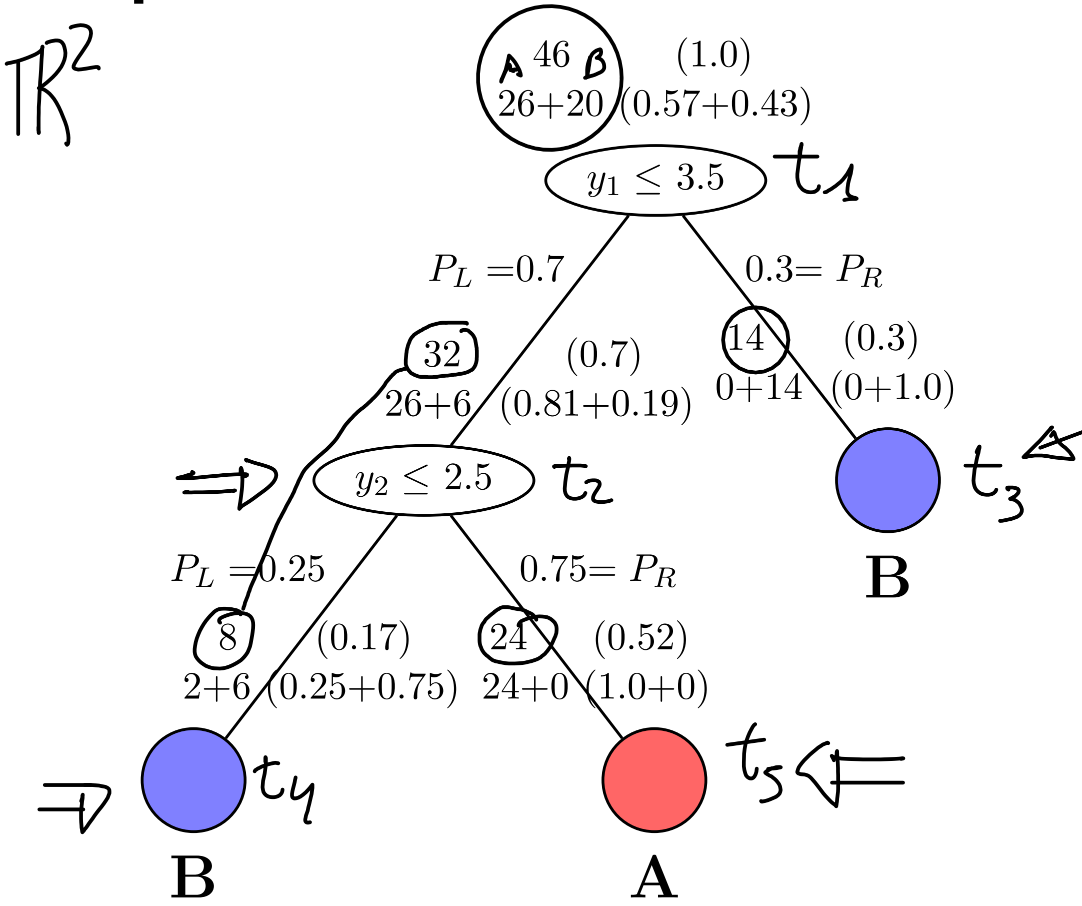
Todo nodo no terminal t verifica las siguients propiedades:

$$\boxed{\hat{P}_L(t) + \hat{P}_R(t) \;=\; 1}$$

Ejercicio de cálculo de probabilidades

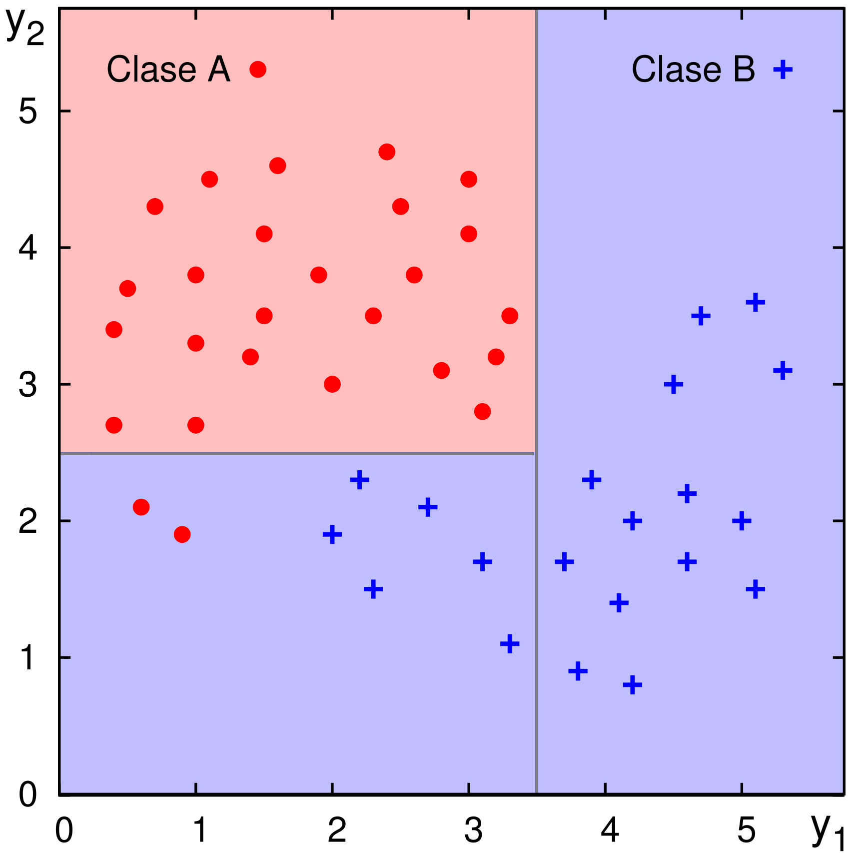


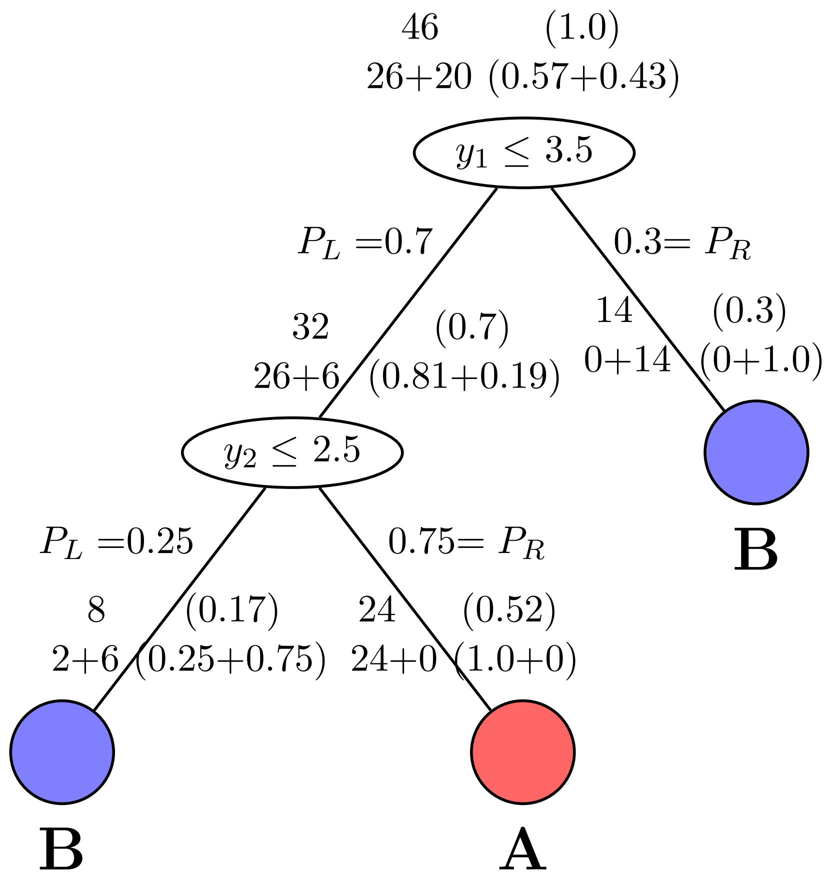
$E=\mathbb{R}^2$



Nodos:	$\hat{P}(t_i)$	$\hat{P}(A \mid t_i)$	$\hat{P}(B \mid t_i)$	$\hat{P}_L(t_i)$	$\hat{P}_R(t_i)$
t_1 (raiz)	—	26/46	20/46	32/46	14/46
t_2 (interno)	—	26/32	6/32	8/32	24/32
t_3 (hoja B)	14/46				
t_4 (hoja B)	8/46				
t_5 (hoja A)	24/46				

Solución al cálculo de probabilidades





Nodos:	$\hat{P}(t_i)$	$\hat{P}(A \mid t_i)$	$\hat{P}(B \mid t_i)$	$\hat{P}_L(t_i)$	$\hat{P}_R(t_i)$
t_1 (raiz)	46/46	26/46	20/46	32/46	14/46
t_2 (interno)	32/46	26/32	6/32	8/32	24/32
t_3 (hoja B)	14/46	0/14	14/14	–	–
t_4 (hoja B)	8/32	2/8	6/8	–	–
t_5 (hoja A)	24/32	24/24	0/24	–	–

Índice

Aprendizaje de un ADC a partir de una muestra de aprendizaje

Parámetros de un algoritmo de aprendizaje de un ADC:

1. Criterio de partición y evaluación de partición:
   - Condiciones o "preguntas" ("splits"): ¿$y \in B$?, $B \subseteq E$ ← ¿$y_j \leq r$?
   - Evaluación y optimización de la calidad de una partición

2. Criterio de *pureza* de un nodo para detener proceso de partición

3. Criterio para asignar una etiqueta de clase a un nodo *terminal*

Conjunto de preguntas admisibles para formar particiones

$s=(j=1, r=2)$

$y_1 \leq 2$

$\{2\}$ $\{5,6\}$

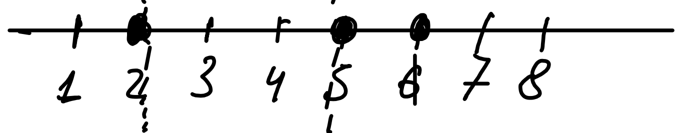
- Cada partición involucra a una única componente j de E, $1 \leq j \leq D$
- Las preguntas ¿$\boldsymbol{y} \in B$? son de la forma: ¿$y_j \leq r$?, $r \in \mathbb{R}$

$y_1 \leq 5$, $y_1 \leq 6$

$y_1 \leq 2$

$y \in \mathbb{R}$

- Una partición o "split" es un par $\boxed{s=(j, r)}$



- Los datos son finitos, por tanto sólo hay un número finito de particiones
- Para un nodo t con $N(t)$ elementos:
  - Hay que explorar cada una de las componentes j, $1 \leq j \leq D$, de E
  - Para cada j, hay que explorar (al menos) $N(t)$ posibles valores de r
- Para cada nodo t, hay que explorar al menos $\mathcal{O}(D \cdot N(t))$ "splits"

$t_1 \to N(t_1)=46$

$\to D=2$ y_1 y_2

$46 \cdot 2 = 92$ particiones

Evaluación de la calidad de una partición

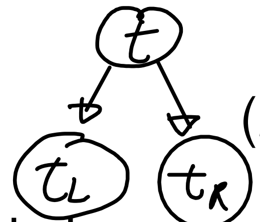
- Para evaluar las particiones posibles se usa el concepto de *"impureza"*
- La impureza de un nodo t, $\mathcal{I}(t)$, se mide en función de las probabilidades estimadas de las clases en t, para lo cual existen varias aproximaciones.
- Una de las más interesantes se basa en el concepto de *entropía* (pág. 16):

$$\mathcal{I}(t) \;=\; -\sum_{c=1}^{C} \hat{P}(c \mid t) \log_2 \hat{P}(c \mid t) \;=\; -\sum_{c=1}^{C} \frac{N_c(t)}{N(t)} \log_2 \frac{N_c(t)}{N(t)} \qquad (1)$$

  Otras definiciones de $\mathcal{I}(t)$: *índice Gini* y *probabilidad de error* (ver [1,2,3])

- La calidad de una partición del nodo t mediante un *"split"* $s = (j, r)$, se mide mediante el *decremento de impureza*:

$$\boxed{\Delta\mathcal{I}(j, r, t)} \stackrel{\text{def}}{=} \mathcal{I}(t) - \hat{P}_L(t)\mathcal{I}(t_L) - \hat{P}_R(t)\mathcal{I}(t_R) \qquad (2)$$



- La mejor partición es aquella que produce mayor decremento de impureza:

$$(j^\star, r^\star) \;=\; \boxed{\underset{\substack{1 \le j \le D \\ -\infty < r < +\infty}}{\operatorname{argmax}}} \boxed{\Delta\mathcal{I}(j, r, t)} \qquad (3)$$

$\Delta I(j=1, r=3.5, t=t_1)$ $y_1 \le 3.5$

$N(t)$

$D=2$ $j=\{1,2\}$

$1 \le j \le D=2$

Entropía

- Mide la cantidad de información asociada a una decisión $k-$aria:

$$H = -\sum_{i=1}^{k} P_i \log_2 P_i \qquad (0 \log 0 \stackrel{\text{def}}{=} 0)$$

  - La unidad es el *bit*: información asociada a tomar una decisión *binaria* en la que las dos alternativas son equiprobables.
  - El valor mínimo es 0 y corresponde a una decisión en la que solo hay una alternativa posible.
  - El valor máximo es $+\infty$ que se da para decisiones k-arias equiprobables con $k \to \infty$:

- Ejemplos:

Si $P_1 = P_2 = 1/2$, $\quad H = -(0.5(0-1) + 0.5(0-1)) = \underline{1}$

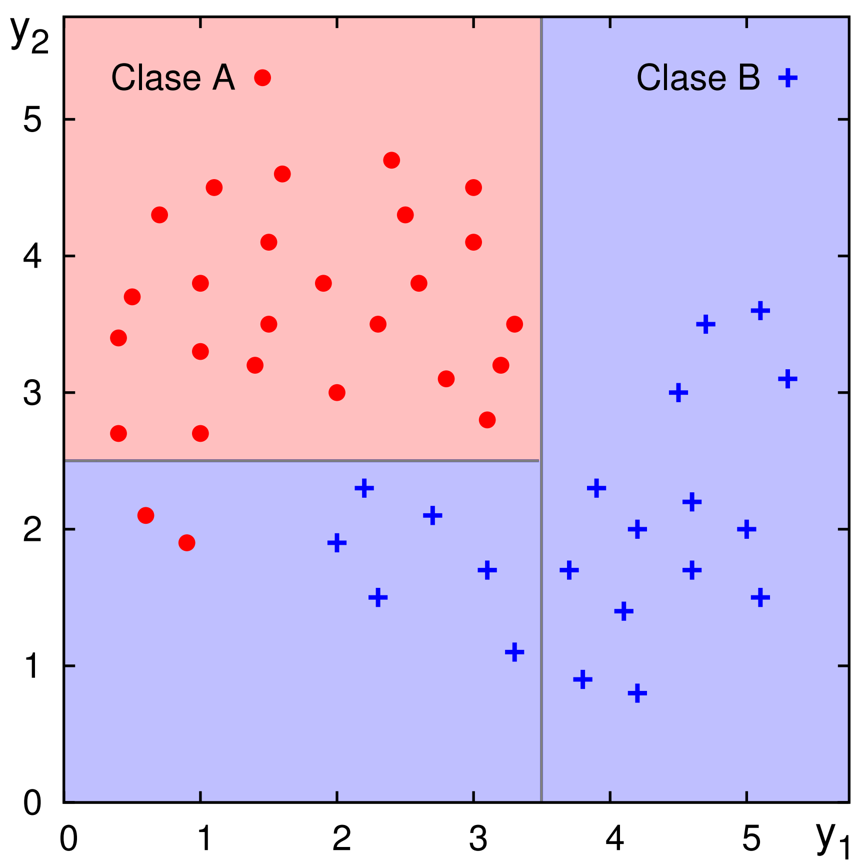
Si $\underline{P_1 = 1}$, $\underline{P_2 = 0}$, $\quad H = -1 \cdot 0 + 0 = \underline{0}$

Si $\underline{P_i = 1/k,\ 1 \le i \le k}$, $\quad H = \underline{log_2 k}$; $\underline{H \to \infty}$ si $\underline{k \to \infty}$

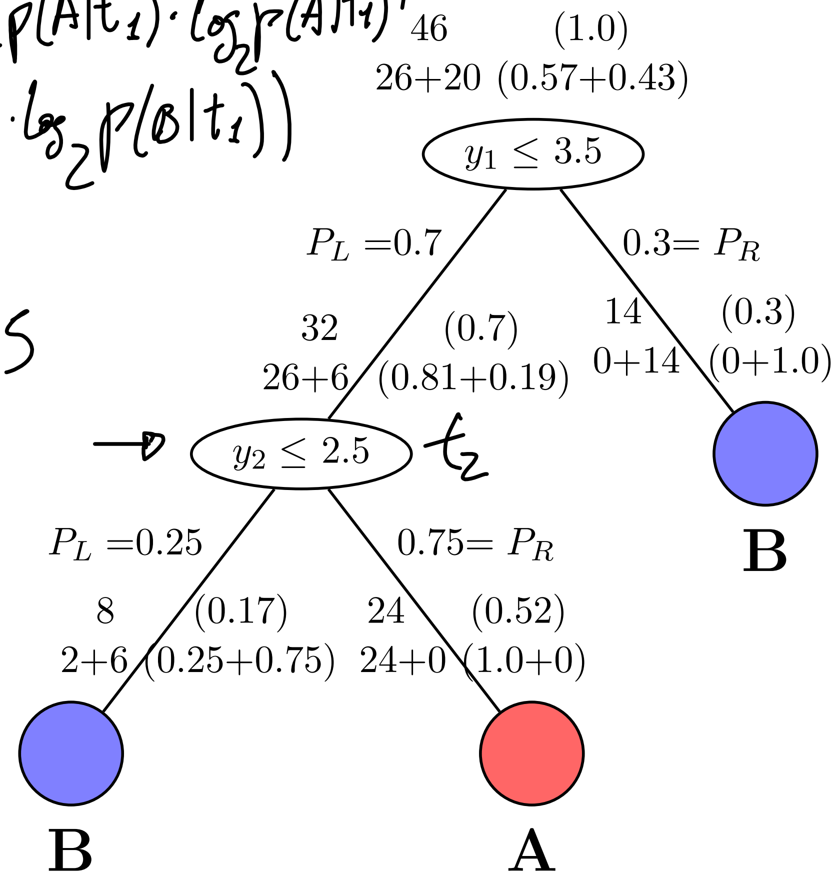
Ejercicio: según Eq.(1), calcular $\mathcal{I}(t)\ \forall t$ en el árbol de la página 5.

Ejercicio de cálculo de impureza

$$I(t_1) = -\left(p(A|t_1)\cdot \log_2 p(A|t_1) + p(B|t_1)\cdot \log_2 p(B|t_1)\right)$$



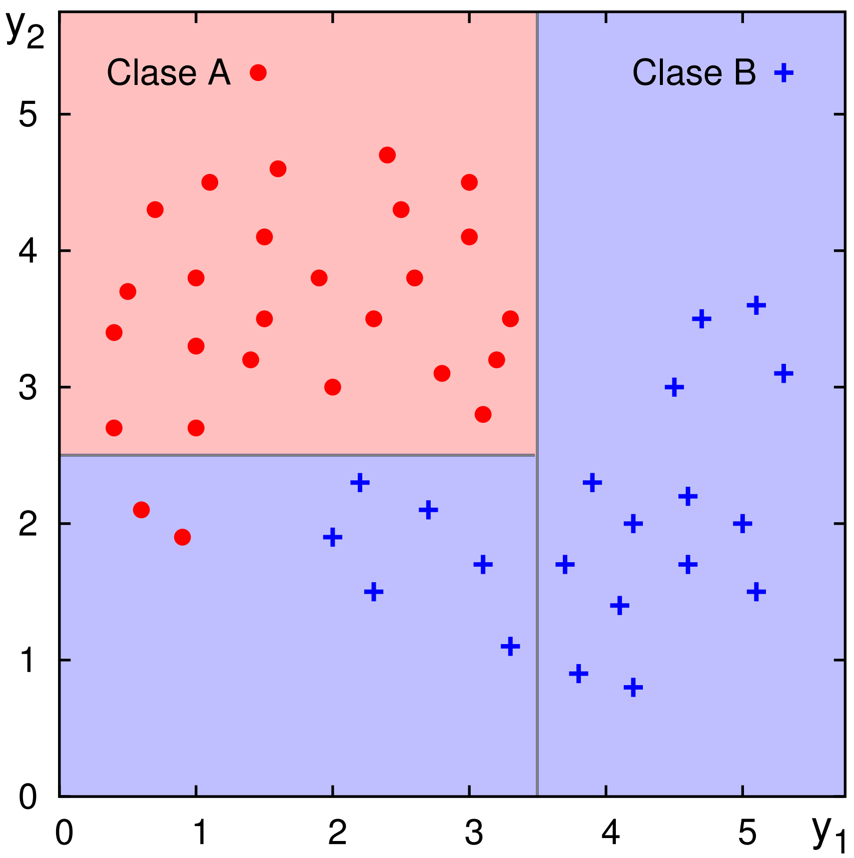
$y_2 \leq 2.5$

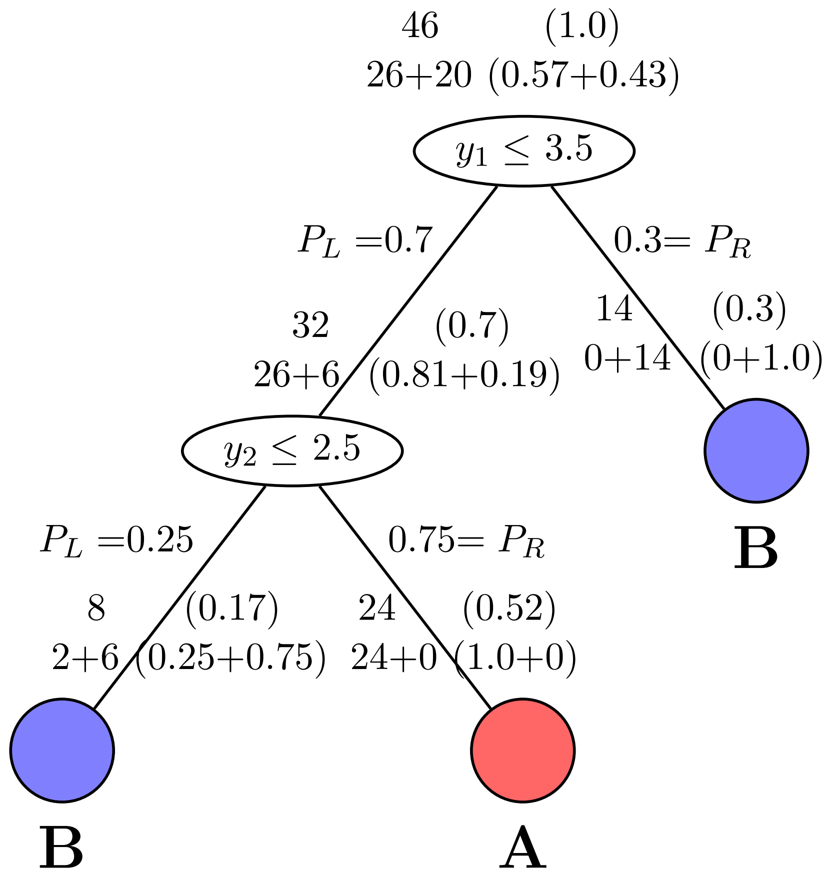


Nodos:	$\hat{P}(t_i)$	$\hat{P}(A \mid t_i)$	$\hat{P}(B \mid t_i)$	$\hat{P}_L(t_i)$	$\hat{P}_R(t_i)$	$\mathcal{I}(t_i)$
t_1 (raiz)	46/46	26/46	20/46	32/46	14/46	
t_2 (interno)	32/46	26/32	6/32	8/32	24/32	
t_3 (hoja B)	14/46	0/14	14/14	–	–	
t_4 (hoja B)	8/32	2/8	6/8	–	–	
t_5 (hoja A)	24/32	24/24	0/24	–	–	

$$I(t_2) = -\left(P(A|t_2)\cdot \log_2 P(A|t_2) + P(B|t_2)\cdot \log_2 P(B|t_2)\right) = -\left(\frac{26}{32}\cdot \log_2 \frac{26}{32} + \frac{6}{32}\cdot \log_2 \frac{6}{32}\right) = 0.6$$

Solución al cálculo de impureza





Nodos:	$\hat{P}(t_i)$	$\hat{P}(A \mid t_i)$	$\hat{P}(B \mid t_i)$	$\hat{P}_L(t_i)$	$\hat{P}_R(t_i)$	$\mathcal{I}(t_i)$
t_1 (raiz)	46/46	26/46	20/46	32/46	14/46	**0.988**
t_2 (interno)	32/46	26/32	6/32	8/32	24/32	**0.695**
t_3 (hoja B)	14/46	0/14	14/14	–	–	**0.000**
t_4 (hoja B)	8/32	2/8	6/8	–	–	**0.811**
t_5 (hoja A)	24/32	24/24	0/24	–	–	**0.000**

Ejercicio (para hacer en clase)

Con respecto al ejemplo de la página 5:

- Calcular el decremento de impureza que se produce al dividir cada uno de los 2 nodos *no* terminales según Eq (2)

- Las dos particiones que se muestran en este ejemplo son solo ejemplos ilustrativos basados en pura intuición geométrica (es decir, *no* son el resultado de la optimización de ninguna expresión de impureza).

  Según la Eq. (3), analizar la segunda partición (la que en el ejemplo se resuelve con $s = (2, 2.5)$; es decir, $y_2 \leq 2.5$), y determinar si alguna de las siguientes particiones es mejor para ese nodo: $(y_1 \leq 1.95)$, $(y_2 \leq 1.8)$

Nodos:	$\hat{P}(t_i)$	$\hat{P}(A \mid t_i)$	$\hat{P}(B \mid t_i)$	$\hat{P}_L(t_i)$	$\hat{P}_R(t_i)$	$\mathcal{I}(t_i)$
t_1 (raiz)	46/46	26/46	20/46	32/46	14/46	**0.988**
t_2 (interno)	32/46	26/32	6/32	8/32	24/32	**0.695**
t_3 (hoja B)	14/46	0/14	14/14	–	–	**0.000**
t_4 (hoja B)	8/32	2/8	6/8	–	–	**0.811**
t_5 (hoja A)	24/32	24/24	0/24	–	–	**0.000**

Solución al ejercicio

- Decrementos de impureza para t_1 y t_2:

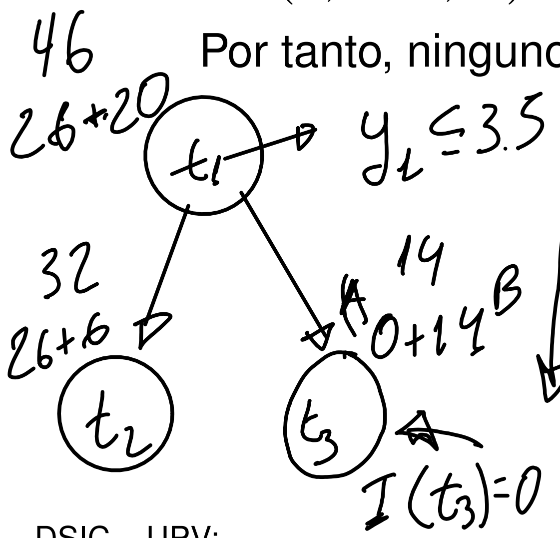
Nodos:	$\hat{P}(t_i)$	$\hat{P}(A \mid t_i)$	$\hat{P}(B \mid t_i)$	$\hat{P}_L(t_i)$	$\hat{P}_R(t_i)$	$\mathcal{I}(t_i)$	$\Delta\mathcal{I}(t_i)$
t_1 (raiz)	46/46	26/46	20/46	32/46	14/46	**0.504**	
t_2 (interno)	32/46	26/32	6/32	8/32	24/32	**0.492**	
t_3 (hoja B)	14/46	0/14	14/14	–	–	–	
t_4 (hoja B)	8/32	2/8	6/8	–	–	–	
t_5 (hoja A)	24/32	24/24	0/24	–	–	–	

- *Splits* alternativos en t_2:

$$(y_1 \leq 1.95) : \mathcal{I}(t_L) = 0,\ \mathcal{I}(t_R) = -(11/17)\log(11/17) - (6/17)\log(6/17) = 0.937$$
$$(y_2 \leq 1.80) : \mathcal{I}(t_L) = 0,\ \mathcal{I}(t_R) = -(26/29)\log(26/29) - (3/29)\log(3/29) = 0.480$$

$$\Delta\mathcal{I}(1, 1.95, t_2) = 0.695 - 0 - (17/32)\cdot 0.937 = 0.197 < 0.492$$
$$\Delta\mathcal{I}(2, 1.80, t_2) = 0.695 - 0 - (29/32)\cdot 0.480 = 0.260 < 0.492$$

Por tanto, ninguno de estos *splits* habría sido mejor que $(y_2 \leq 2.5)$.



$\Delta I(j=1, r=3.5, t=t_1) = I(t_1) - P_L(t_1)\cdot I(t_2) - P_R(t_1)\cdot I(t_3)$

$0.504 - \frac{32}{46}\cdot 0.492 - \frac{14}{46}\cdot 0 = 0.16$

$I(t_3) = -\left(\frac{0}{14}\log_2\frac{0}{14} + \frac{14}{14}\cdot\log_2\frac{14}{14}\right) = 0$

$0\cdot\log 0 = 0$

Criterios de suficiente “pureza” en nodos terminales

- Criterio de parada de particionamiento, cuando máximo decremento de impureza posible es demasiado pequeño:

0.1

$$\boxed{\max_{\substack{1\leq j\leq D \\ -\infty<r<+\infty}} \Delta\mathcal{I}(j,r,t)} < \epsilon \tag{4}$$

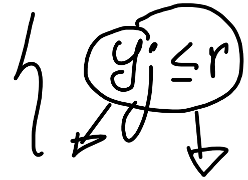
donde ϵ es una constante pequeña a determinar empíricamente.

- Otro criterio es exigir que los nodos terminales sean totalmente puros
- Este último criterio tiene problemas de generalización y árboles grandes

Asignación de etiquetas de clase a nodos terminales

- Criterio simple y eficaz: asignar a cada nodo terminal la clase mayoritaria en sus elementos:

$$c^\star(t) = \operatorname*{argmax}_{1\leq c\leq C} \underline{\hat{P}(c \mid t)}, \qquad \forall t \in \tilde{T} \tag{5}$$

Árbol 

A

$y_{j^*} \le r^*$

$\delta = \Delta I(j^*, r^*, t)$

Algoritmo ADC

Árbol CreaNodo(*clase* c, *componente* j, *umbral* u, *nodos* t_L, t_R) //crea un árbol (↑nodo)

Árbol ADC(*muestra* $\mathcal{Y} \equiv (\boldsymbol{y}_1, c_1), \ldots, (\boldsymbol{y}_n, c_n)$) { // aprende un Árbol de Clasificación

$(j^\star, r^\star, \delta) =$ MejorPartición($\mathcal{Y}$) // según Eq. (1,2,3); δ es el decremento de impureza

if ($\delta < \epsilon$) { Paramos de particionar // si δ es demasiado pequeño – según Eq. (4)

$c =$ ClaseMayoritaria($\mathcal{Y}$) $c = \operatorname{argmax}_c P(c \mid t)$ // según Eq. (5)

return CreaNodo(c, –, –, NULL, NULL) // crea nodo terminal y le asigna la clase c

} ***else*** {

$\mathcal{Y}_L = \mathcal{Y}_R = \emptyset$ conjuntos de datos para hijo izq y dcho // $\emptyset$ es el conjunto vacío

$\forall(\boldsymbol{y}, c) \in \mathcal{Y}$ { // realiza la partición en función de $j^\star, r^\star$

if $(y_{j^\star} \le r^\star)$ $\mathcal{Y}_L = \mathcal{Y}_L \cup \{(\boldsymbol{y}, c)\}$

else /* $(y_{j^\star} > r^\star)$ */ $\mathcal{Y}_R = \mathcal{Y}_R \cup \{(\boldsymbol{y}, c)\}$

}

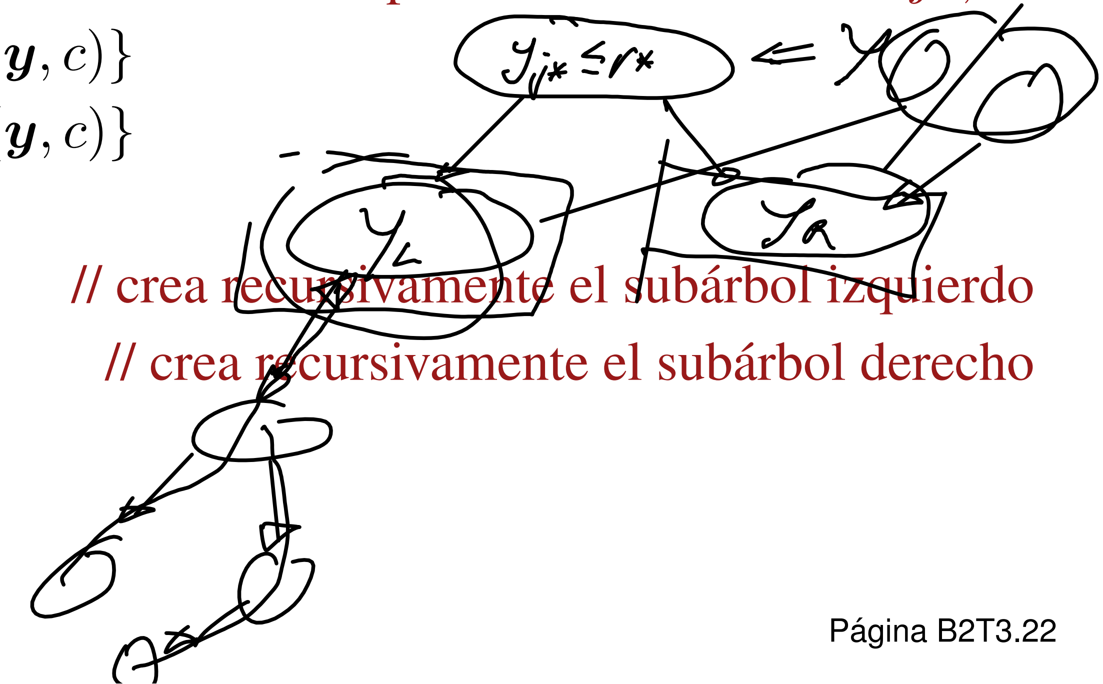
$t_L =$ ADC($\mathcal{Y}_L$) // crea recursivamente el subárbol izquierdo

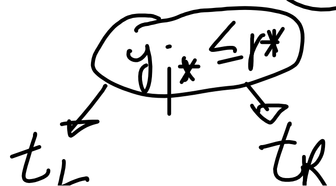
$t_R =$ ADC($\mathcal{Y}_R$) // crea recursivamente el subárbol derecho

return CreaNodo($0, j^\star, r^\star, t_L, t_R$)

}

}





Estimación por resustitución del error de clasificación

- Según la *teoría de la decisión estadística*, la probabilidad de error de un nodo (terminal) t, estimada por *resustitución*, es:

$$\hat{P}_e(t) \;=\; 1 - \max_{1 \leq c \leq C} \hat{P}(c \mid t)$$

  Y para un árbol T:

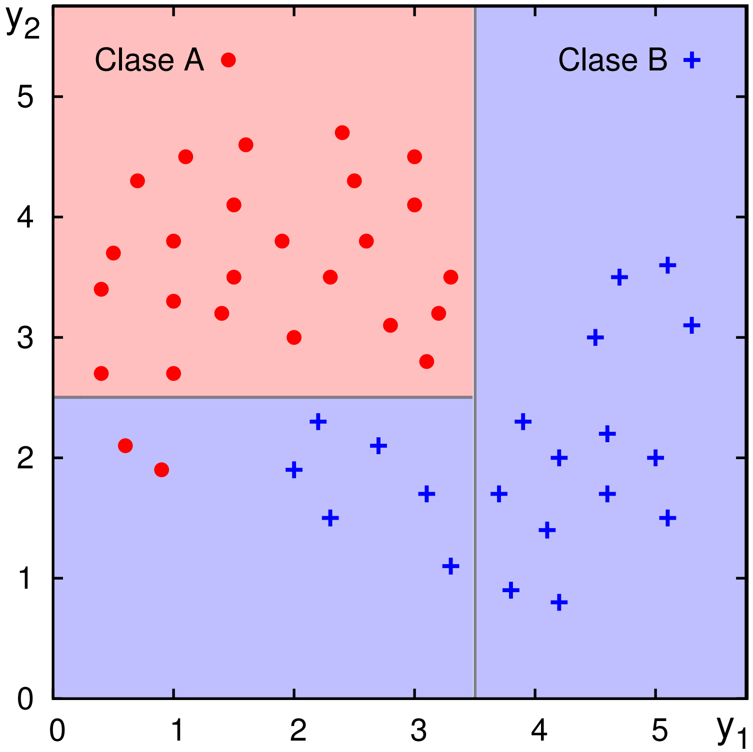
$$\hat{P}_e(T) \;=\; \sum_{t \in \tilde{T}} \underline{\hat{P}(t)}\,\underline{\hat{P}_e(t)}$$

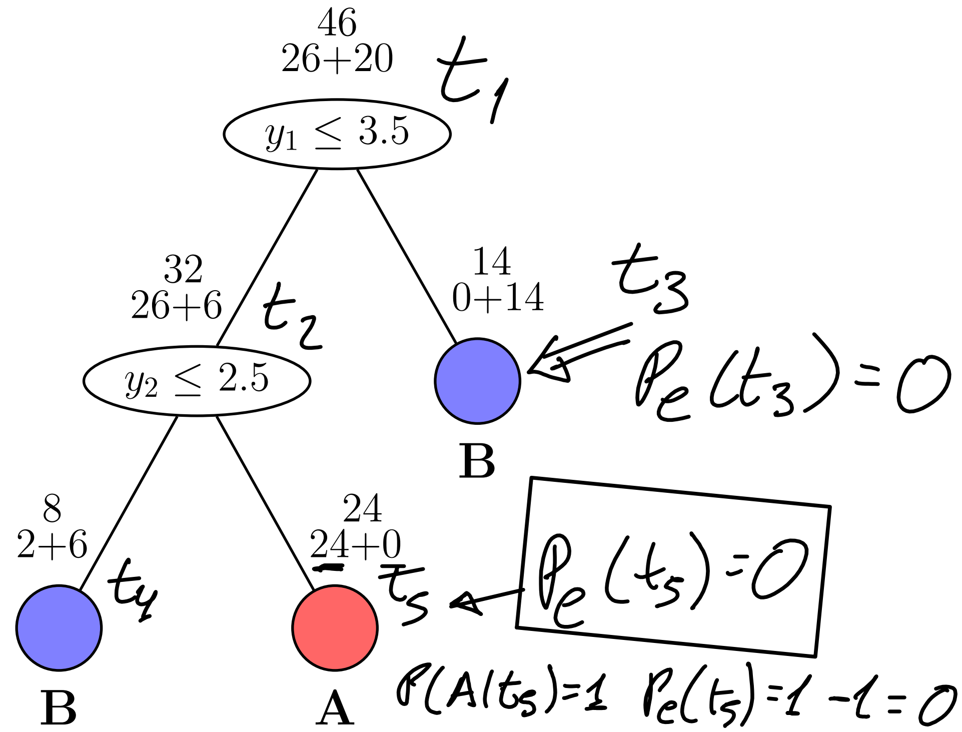
- Si se hace crecer el árbol hasta que los nodos terminales sean totalmente puros, el error estimado será nulo, ya que en este caso:

$$\hat{P}_e(t) = 0 \quad \forall t \in \tilde{T}.$$

- Esto conlleva un *sobreaprendizaje* que generalmente no es desable $\Rightarrow$ esencialmente el árbol se convierte en mero almacén de la muestra de aprendizaje, sin capacidad de generalización ante nuevos datos

Estimación por resustitución del error de clasificación



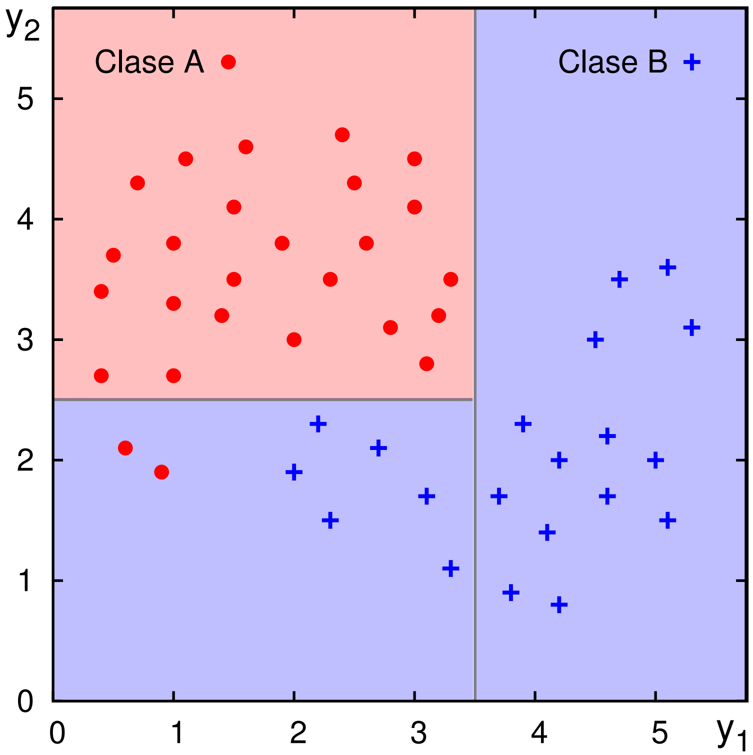


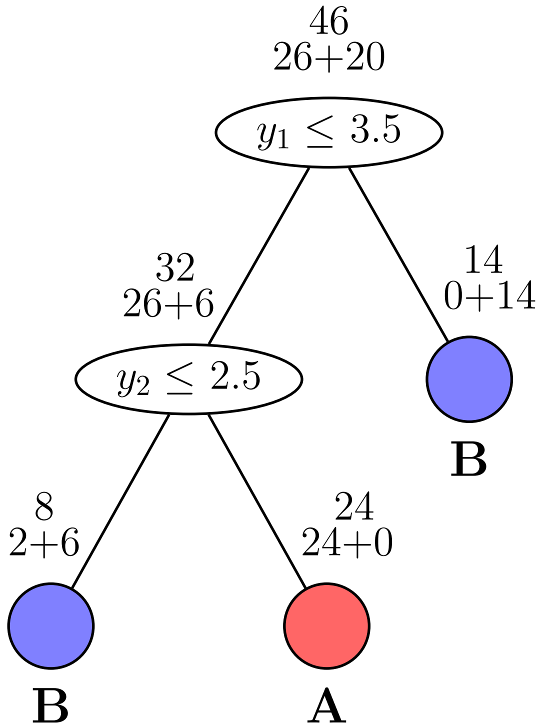
$$P_e(t_3) = 1 - \max_{c} P(c|t_3) = 1 - \max(P(A|t_3), P(B|t_3)) = 1-1 = 0$$

$$P_e(t_4) = 1 - \frac{6}{8} = 2/8 = 1/4$$

$$P_e(T) = \underline{P(t_3)}\cdot P_e(t_3) + P(t_4)\cdot P_e(t_4) + P(t_5)\cdot P_e(t_5) = \frac{14}{46}\cdot 0 + \frac{8}{46}\cdot\frac{1}{4} + \frac{24}{46}\cdot 0 = 0.0435$$

Estimación por resustitución del error de clasificación





En el ejemplo de la página 5, (3 nodos terminales, 2 totalmente puros):

$$\hat{P}_e(T) = \hat{P}(t_3)\hat{P}_e(t_3) + \hat{P}(t_4)\hat{P}_e(t_4) + \hat{P}(t_5)\hat{P}_e(t_5)$$

$$= \frac{14}{46} \cdot 0 + \frac{8}{46} \cdot \frac{2}{8} + \frac{24}{46} \cdot 0 \;=\; \frac{2}{46} \;\approx\; 0.0435 \;\rightarrow\; 4.35\,\%$$

Algoritmo ADC: Ejercicio en clase

En un problema de clasificación en 3 clases, para objetos representados mediante vectores de características bidimensionales tenemos la siguiente muestra:

C

	Entrenamiento					Test			
n	1	2	3	4	5	6	7	8	9
x_{n1}	1	1	2	2	3	3	3	0	0
x_{n2}	0	1	0	1	0	1	-1	1	0
c_n	A	B	B	B	C	B	C	B	A

1. Realiza una traza de ejecución del algoritmo ADC sobre el conjunto de entrenamiento siendo $\epsilon = 0$. Muestra el árbol resultante, los valores de impureza de los nodos generados y los decrementos de impureza de las particiones obtenidas.

2. Representa gráficamente las particiones obtenidas y las regiones de decisión tras la ejecución del algoritmo ADC, así como las muestras de entrenamiento.

3. Calcula la probabilidad de error en test con intervalos de confianza al 95 % y discute el resultado.

4. Muestra el árbol resultante si $\underline{\epsilon = 0.5}$ y calcula la probabilidad de error en test en este caso.

Ejercicio de seguimiento: Algoritmo ADC

En un problema de clasificación en 3 clases, para objetos representados mediante vectores de características tridimensionales tenemos la siguiente muestra:

	Entrenamiento									Test					
n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_{n1}	0	1	0	2	0	1	1	2	2	1	0	0	1	1	2
x_{n2}	0	1	1	1	0	1	0	1	1	0	1	0	0	1	0
x_{n3}	0	0	1	0	2	3	2	2	3	0	2	1	3	1	2
c_n	A	A	A	A	B	B	C	C	C	A	A	B	B	C	C

1. A partir del conjunto de entrenamiento, enumera el conjunto de particiones posibles para el nodo raíz en las que ambos nodos hijos tienen al menos un elemento.

2. Calcula la mejor partición inicial (nodo raíz) de acuerdo al decremento de impureza.

3. Calcula la probabilidad de error en test tras aplicar el algoritmo ADC con $\epsilon = 0.35$

4. Considera sólo las muestras de entrenamiento de las clases B y C. ¿Es posible definir una única partición con error de resustitución igual a cero? ¿Por qué?

5. En el caso del apartado anterior, ¿crees que el algoritmo Perceptron conseguiría un error de resustitución igual a cero? ¿Por qué?

Índice

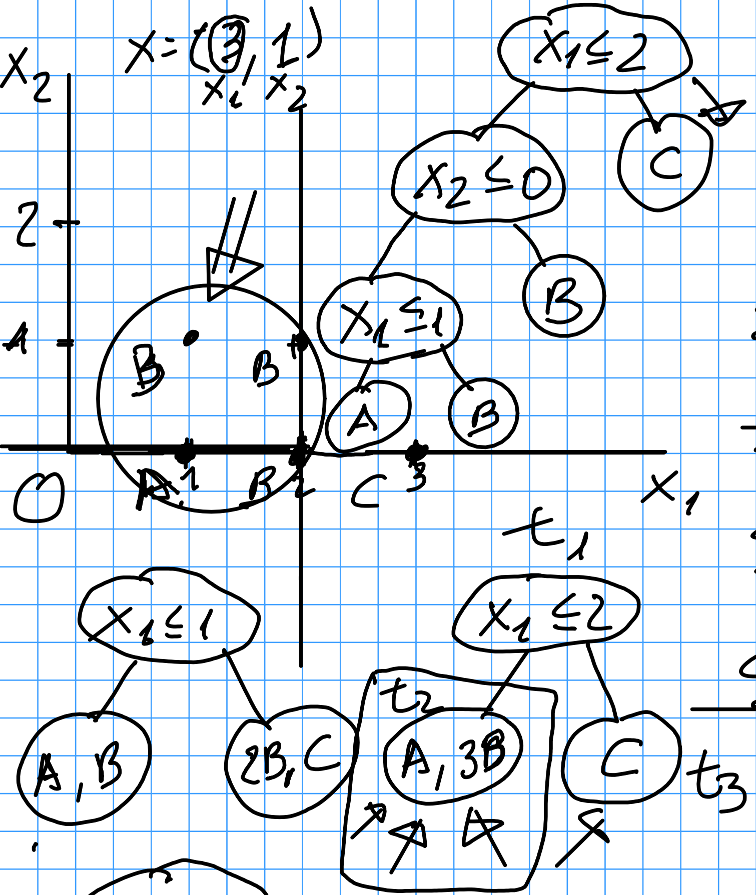
Bibliografía

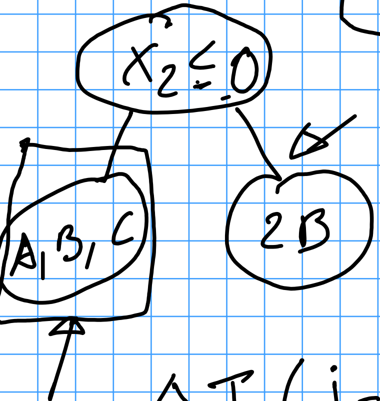
[1] R.O. Duda, D.G. Stork, P.E. Hart. Pattern Classification. Wiley, 2001.

[2] A. R. Webb, K. D. Copsey. Statistical Pattern Recognition. Wiley, tercera ed., 2011.

[3] Classification and Regression Trees by L. Breiman, J.H. Friedman, R.A. Olshen y C.J. Stone. Chapman & Hall, 1984.

El material de este tema se basa principalmente en [1] y [2].





$x_j \le r$

$S = (1, 1) \quad x_1 \le 1$

$S = (1, 2) \quad x_1 \le 2$

$\cancel{S = (1, 3) \quad x_1 \le 3}$

$S = (2, 0) \quad x_2 \le 0$

$\cancel{S = (2, 1) \quad x_2 \le 1}$

$$I(t_1) = -\sum_c P(c|t_1) \cdot \log_2 P(c|t_1) =$$

$$= -\big(P(A|t_1) \cdot \log P(A|t_1) + P(B|t_1) \cdot \log P(B|t_1)$$

$$+ P(C|t_1) \cdot \log P(C|t_1)\big) =$$

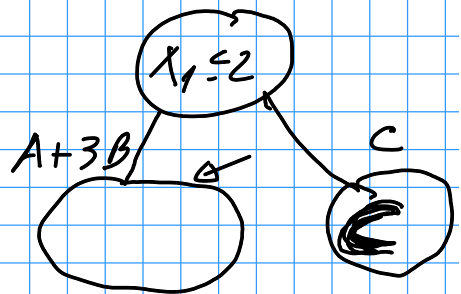
$$= -\left(\frac{1}{5} \cdot \log_2 \frac{1}{5} + \frac{3}{5} \cdot \log_2 \frac{3}{5} + \frac{1}{5} \cdot \log_2 \frac{1}{5}\right)$$

$$= 1.37$$

$$I(t_2) = -\left(\frac{1}{4} \cdot \log \frac{1}{4} + \frac{3}{4} \cdot \log \frac{3}{4}\right) = 0.81 \qquad I(t_3) = 0$$

$$P_L(t_1) = 4/5 \qquad P_R(t_1) = 1/5$$

$$\boxed{S = (1, 2)}$$



$$\Delta I(j=1, r=2, t_1) = I(t_1) - P_L(t_1) \cdot I(t_2) - P_R(t_1) \cdot I(t_3)$$

$$= 1.37 - 4/5 \cdot 0.81 - 1/5 \cdot 0 = 0.72$$

$S = (1, 1)$

$S^* = (2, 0)$