Métodos de Clasificación *sin* Métrica

- *Notas basadas en el curso Reconocimiento de Formas de F.Cortijo, Univ. de Granada*
- *Pattern Classification de Duda, Hart y Storck*
- *The Elements of Statistical Learning de Hastie, Tibshirani y Friedman*
- *Parte del material se extrajo de las notas:Técnicas Supervisadas II: Aproximación no paramétrica de F.Cortijo, Univ. de Granada*

Contenido

- (Resumen) Clase anterior

- Métodos de Clasificación sin Métrica

  → Árboles de Decisión

Repaso

Métodos de Clasificación *sin* Métrica

Métodos de Clasificación sin Métrica

- Datos **nominales** (discretos) sin noción de similitud o distancia
- Escala nominal: conjunto de categorías mutuamente excluyentes y globalmente exhaustivas.
- Ej: Clasificación de frutas
  - Características: color, textura, sabor, tamaño
  - x={rojo, brillante, dulce, pequeño}

Métodos de Clasificación sin Métrica

- Datos **nominales** (discretos) sin noción de similitud o distancia
- Escala nominal: conjunto de categorías mutuamente excluyentes y globalmente exhaustivas.
- Ej: Clasificación de frutas
  - Características: color, textura, sabor, tamaño
  - x={rojo, brillante, dulce, pequeño}
- Características cualitativas (categóricas):
  - Ordinales (existe un orden jerárquico, e.g., grado de educación)
  - Nominales (no existe un orden, e.g., profesión)

Métodos de Clasificación sin Métrica

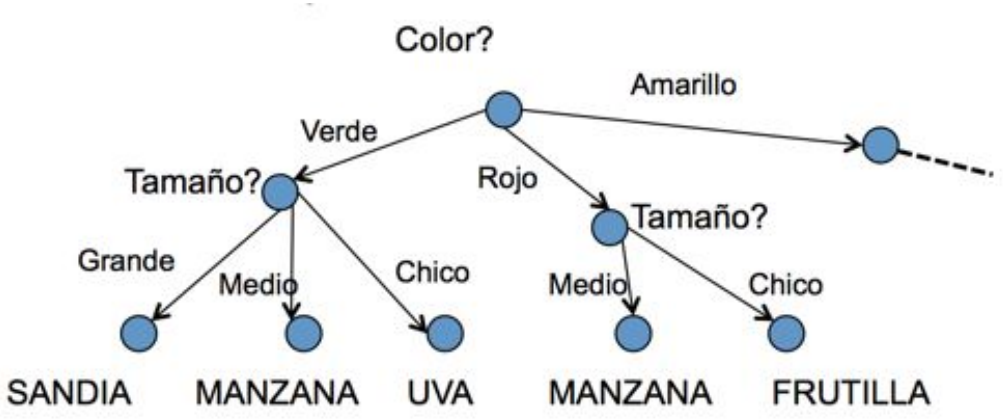
- Datos **nominales** (discretos) sin noción de similitud o distancia
- Escala nominal: conjunto de categorías mutuamente excluyentes y globalmente exhaustivas.
- Ej: Clasificación de frutas
  - Características: color, textura, sabor, tamaño
  - x={rojo, brillante, dulce, pequeño}
- Características cualitativas (categóricas):
  - Ordinales (existe un orden jerárquico, e.g., grado de educación)
  - Nominales (no existe un orden, e.g., profesión)
- **¿Cómo aprender clases usando datos sin métrica?**
- ¿Cuál es la forma más *eficiente* de aprender usando datos nominales para clasificar?

Árboles de Decisión

- Secuencia de preguntas en la que la pregunta siguiente depende de la respuesta de la pregunta actual.
- Particularmente útil para datos sin métrica (usando atributos).

Árboles de Decisión

- Secuencia de preguntas en la que la pregunta siguiente depende de la respuesta de la pregunta actual.
- Particularmente útil para datos sin métrica (usando atributos).
- Clasificador estructura de árbol.
- **Árbol:** consiste en **Nodos interiores** y **Nodos terminales.**
- **Nodo interior**: pregunta sobre un atributo concreto (ramas mutuamente distintas y excluyentes)
- **Nodo terminal u hoja**: asociado a una clase.



Árboles de Decisión

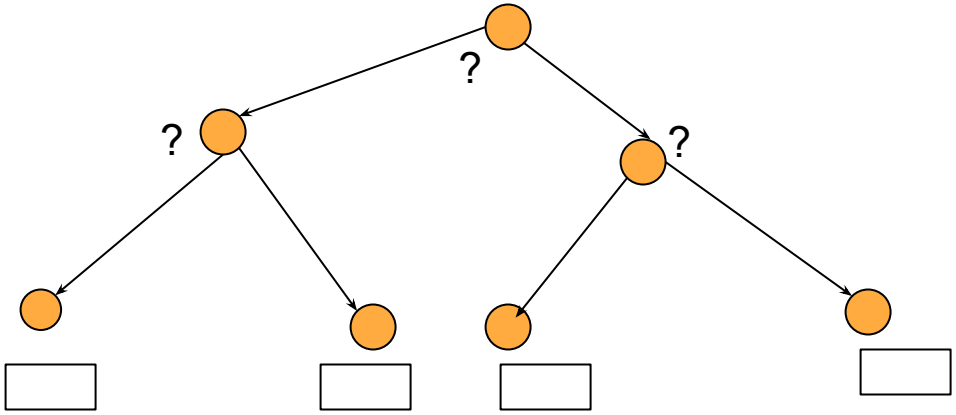
- Aprendizaje. *Construcción del árbol* a partir de un conjunto de muestras etiquetadas.
- Clasificación. Preguntas sobre los valores de los atributos, se comienza por el nodo raíz y se continúa por el camino determinado por las respuestas a las preguntas de los nodos internos, hasta llegar a un nodo hoja. La etiqueta asignada a esta hoja es la que se asignará al patrón a clasificar.

Árboles de Decisión

- Fácilmente **interpretable**: interpretación de las clases en función de los atributos. Ej: MANZANA=(verde y medio) o (rojo y medio). Permite explicar decisiones.
- Adecuados para datos cuantitativos y cualitativos.
- Clasificación es rápida.
- Permite incluir conocimiento a priori de expertos
- Explicitan utilidad de las características.
- Benchmark (referencia) para evaluar desempeño, a veces alcanza el desempeño de clasificadores más complejos/sofisticados.

Construcción de árboles de decisión
(CART, ID3, C4.5)

- CART: (Classification And Regression Trees) Breiman 1984
- ID3, C4.5 Quinlann 1992
- Partimos de patrones etiquetados



Principio fundamental:
Simplicidad del árbol (principio parsimonia o “Navaja de Occam”)
"En igualdad de condiciones, la explicación más sencilla suele ser la más probable."

Construcción de árboles de decisión
(CART, ID3, C4.5)

- Proceso **recursivo**: dados los patrones que llegan a un nodo:
  1. Declarar el nodo terminal (asignamos una clase)
  2. Encontrar una nueva característica y volver a dividir los patrones.

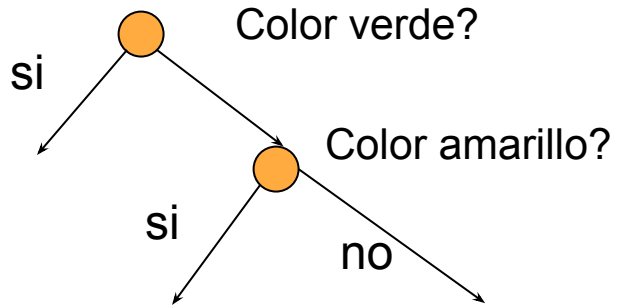
Construcción de árboles de decisión
(CART, ID3, C4.5)

- Proceso **recursivo**: dados los patrones que llegan a un nodo:
  1. Declarar el nodo terminal (asignamos una clase)
  2. Encontrar una nueva característica y volver a dividir los patrones.

- **Preguntas:**
  - ¿Dos ramificaciones o más?
  - ¿Qué atributo se analiza en cada nodo?
  - ¿Cuando un nodo es terminal?
  - ¿Cómo se asigna la clase final?

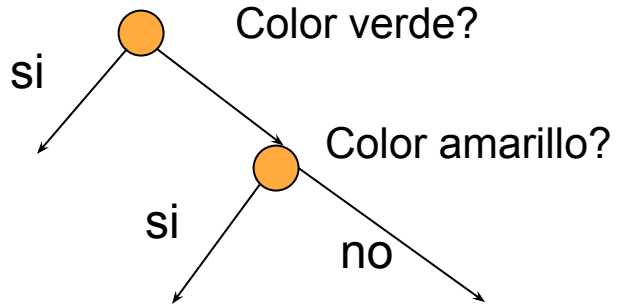
Árboles Binarios

- **Número de ramificaciones:**
  - Binarias
  - No binarias



Árboles Binarios

- **Número de ramificaciones:**
  - Binarias
  - No binarias



- Cualquier decisión (cualquier árbol) puede representarse usando sólo decisiones binarias.

- Nos concentramos en **árboles binarios** (en general son los usados en algoritmos prácticos) :
  - Poder expresivo universal de los árboles binarios.
  - Simplicidad comparativa del entrenamiento.

Construcción del árbol de clasificación

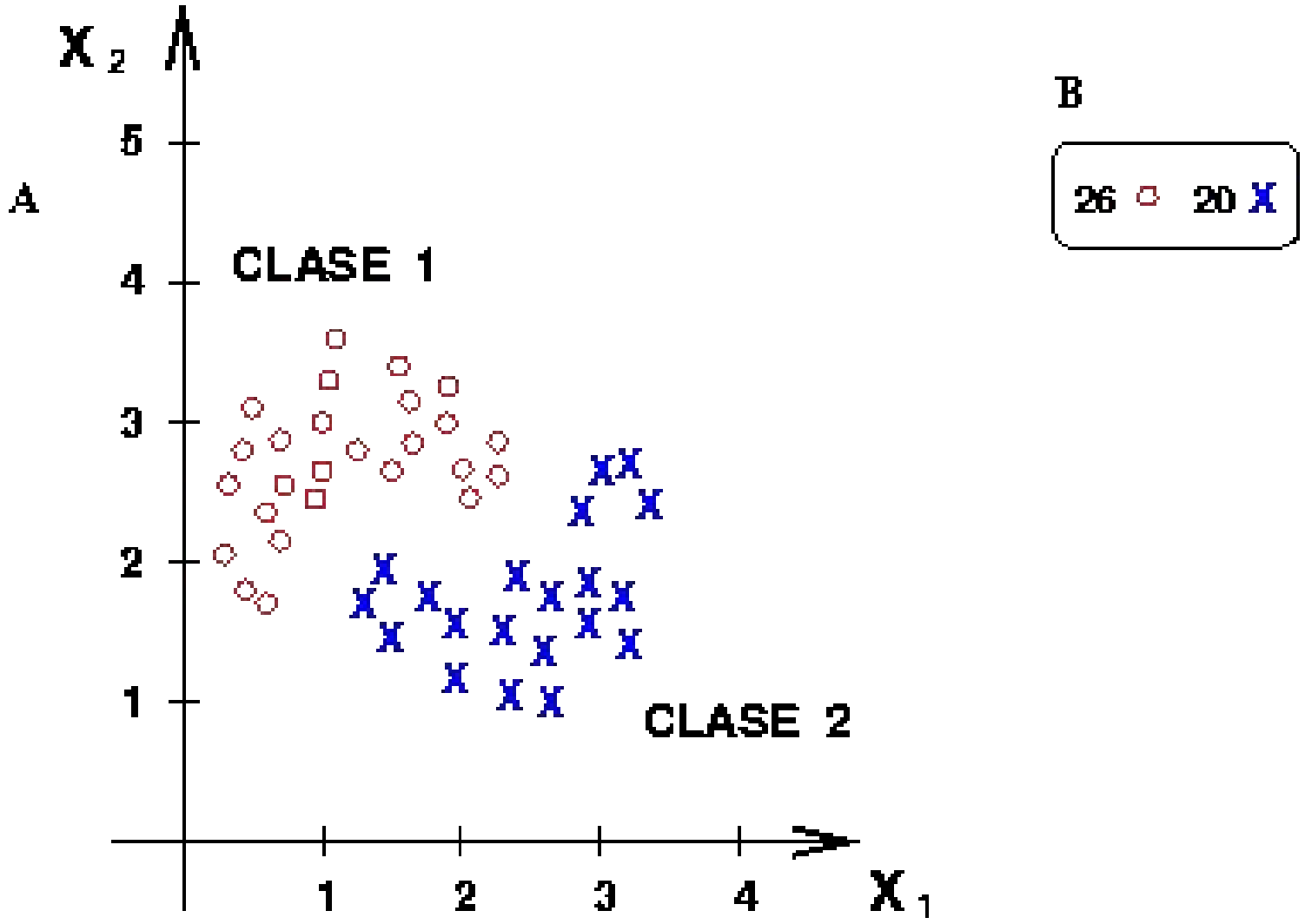
- **Nodo raíz:** Tiene a todos los prototipos

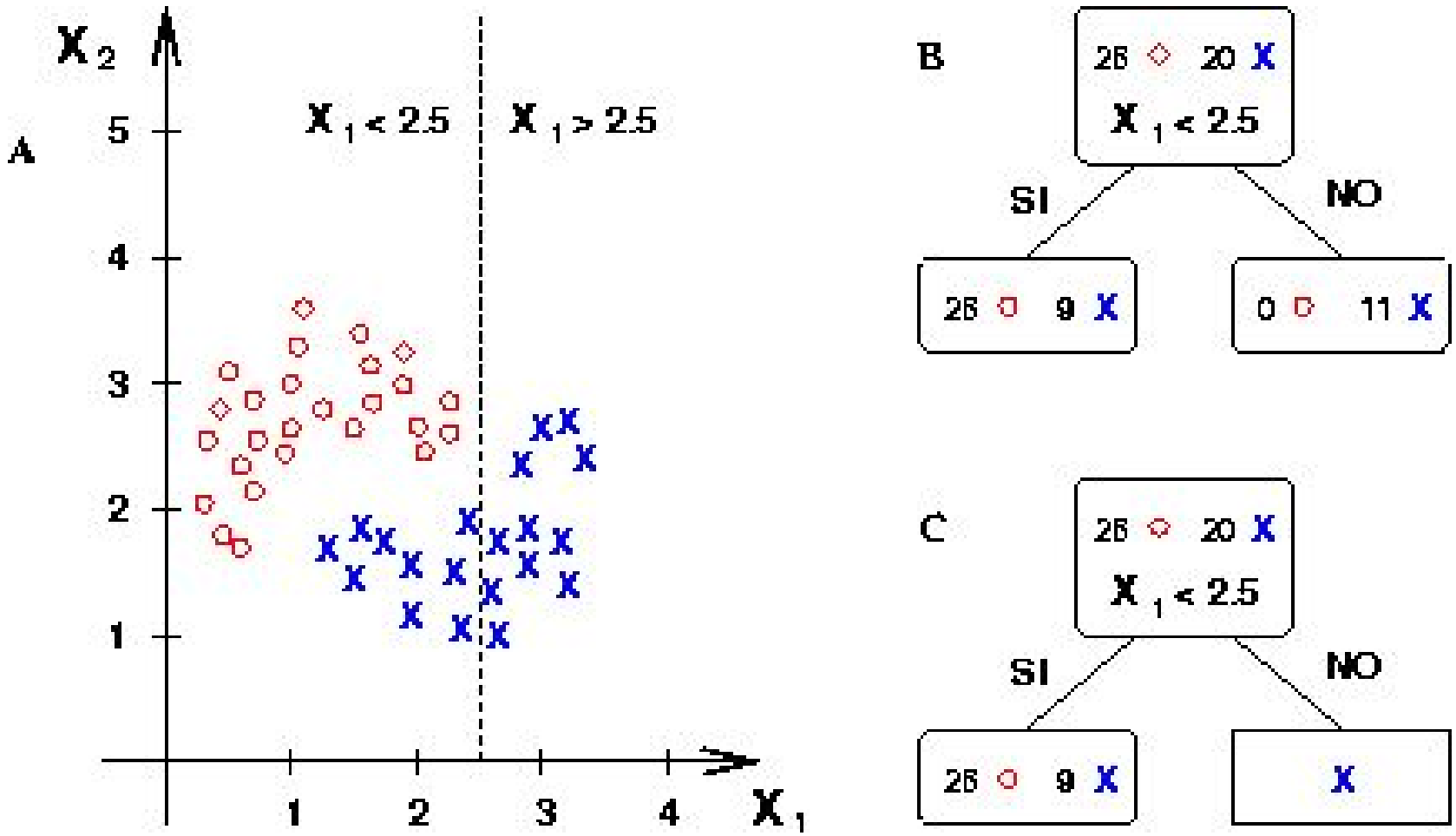
Construcción del árbol de clasificación

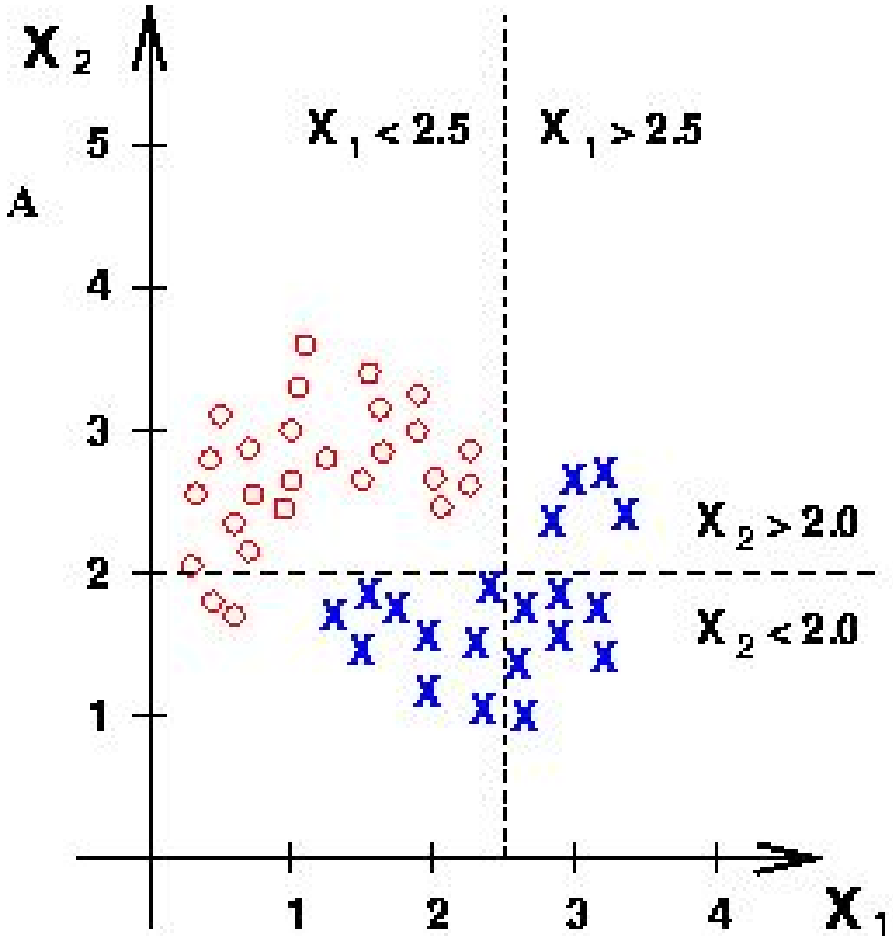
- **Nodo raíz:** Tiene a todos los prototipos
- Se parte el nodo raíz:
  - Dada una característica se elige la partición que separa a los prototipos en clases más puras.
  - Se realiza lo mismo para las otras características.
  - Se **selecciona** la característica y partición que separa mejor las clases (**pureza**)

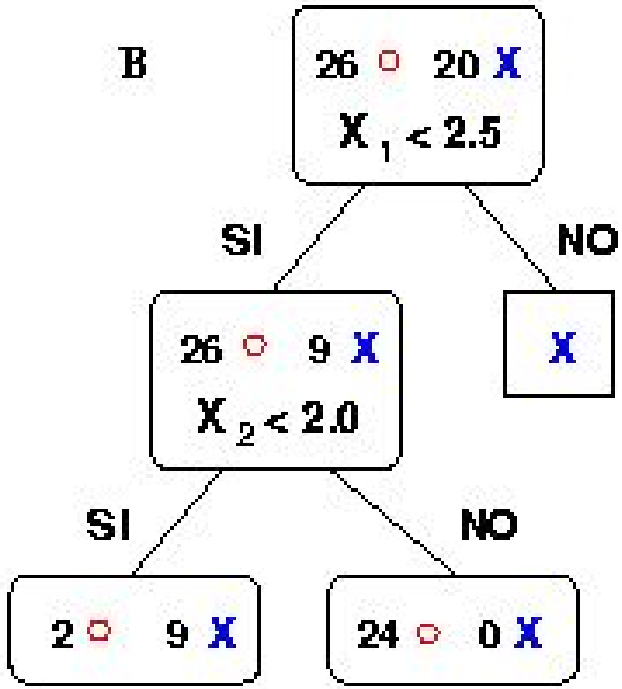
Construcción del árbol de clasificación

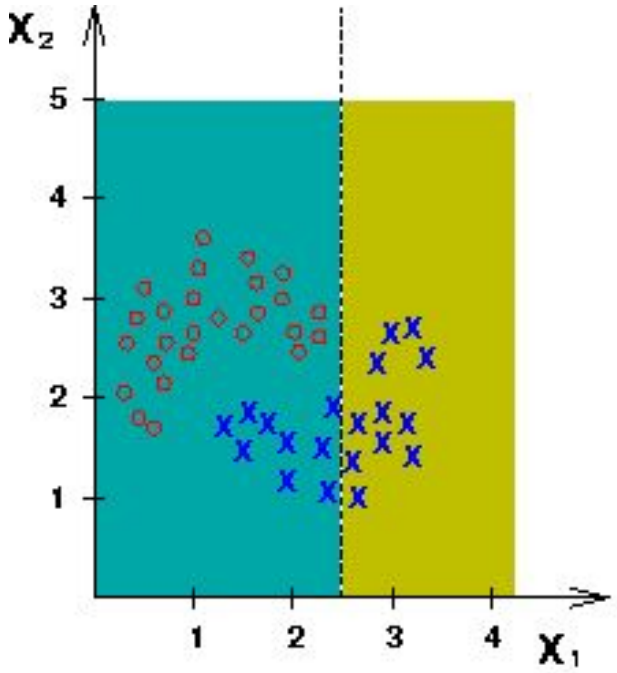
- **Nodo raíz:** Tiene a todos los prototipos
- Se parte el nodo raíz:
  - Dada una característica se elige la partición que separa a los prototipos en clases más puras.
  - Se realiza lo mismo para las otras características.
  - Se **selecciona** la característica y partición que separa mejor las clases (**pureza**)
- Se repite el procedimiento para los nodos hijos hasta llegar a condición de parada (nodo hoja).
- Los prototipos asociados a un nodo hoja se le asigna una *etiqueta* (la de la mayoría).


A
B
26
20
CLASE 1
CLASE 2
X 2
X 1
5
4
3
2
1
1
2
3
4

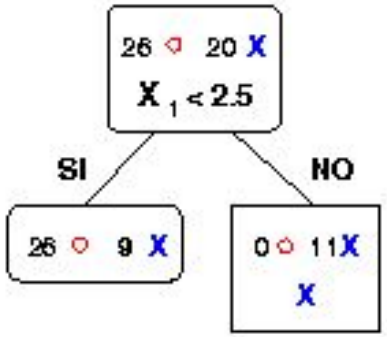

A
X2
X1 < 2.5
X1 > 2.5
5
4
3
2
1
1
2
3
4
X1
B
26 ◇ 20 X
X1 < 2.5
SI
NO
26 ○ 9 X
0 ○ 11 X
C
26 ◇ 20 X
X1 < 2.5
SI
NO
26 ○ 9 X
X

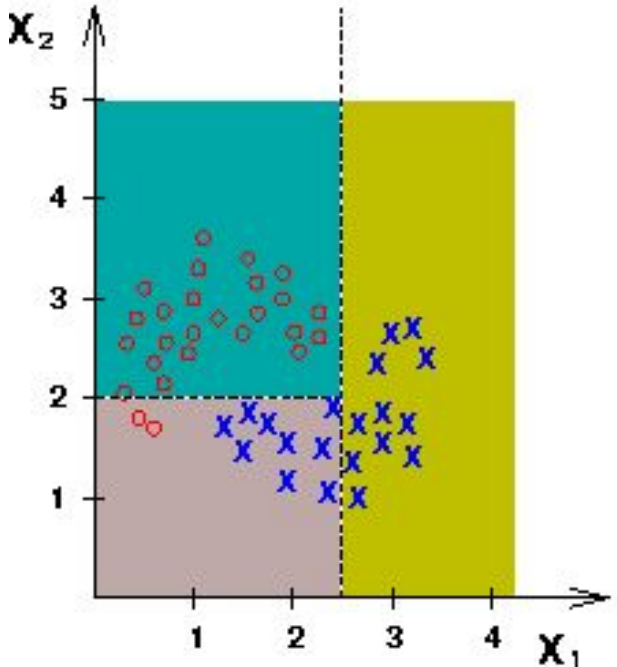

X 2
A
5
4
3
2
1
X 1 < 2.5
X 1 > 2.5
X 2 > 2.0
X 2 < 2.0
1
2
3
4
X 1


B
26 20
X 1 < 2.5
SI
NO
26 9
X 2 < 2.0
X
SI
NO
2 9
24 0

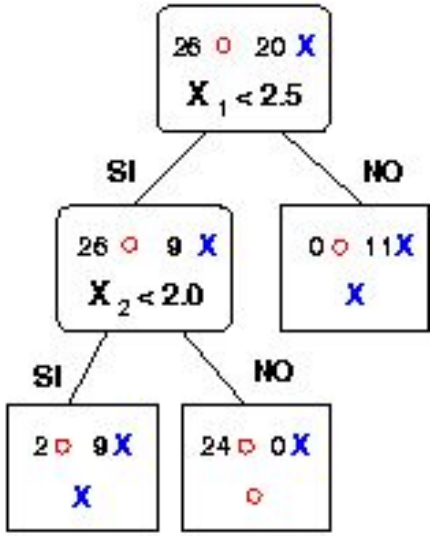

X2
5
4
3
2
1
1
2
3
4
X1

A


26 20
X1 < 2.5
SI
NO
26 9
0 11


X2
5
4
3
2
1
1
2
3
4
X1

B


26 20
X1 < 2.5
SI
NO
26 9
X2 < 2.0
0 11
SI
NO
2 9
24 0

Selección de las particiones

- ¿De qué forma se hacen las **particiones** y se selecciona la mejor de entre las posibles en cada momento?

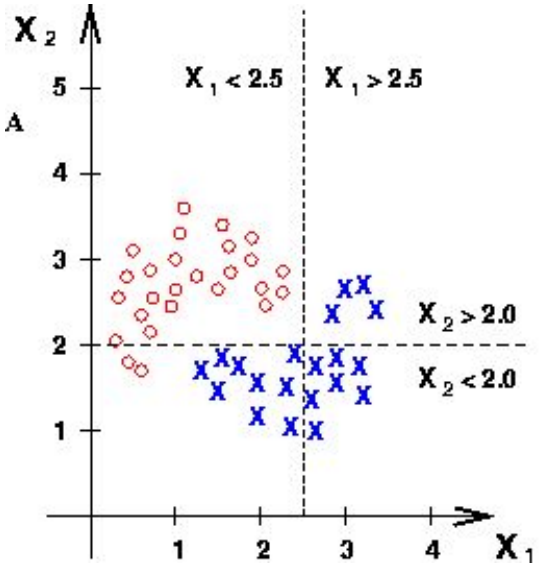
Selección de las particiones

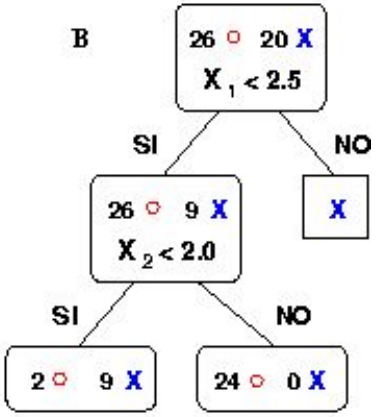
- ¿De qué forma se hacen las **particiones** y se selecciona la mejor de entre las posibles en cada momento?
- **Objetivo:** Incrementar *homogeneidad* de los conjuntos resultantes al particionar (pureza)
- Medida de pureza → Impureza del nodo N: *i(N)*.

  Vamos a analizar distintas opciones para i(N).

Criterios de partición (impureza)

- ***i*(N)** : Impureza de un nodo N: Medida de la homogeneidad de los datos que llegan a ese nodo.
- Es función de: $P(\omega_1), P(\omega_2), \ldots, P(\omega_{n_c})$ $\quad P(\omega_i) = \dfrac{\#_i(N)}{\#(N)}$
- Impureza máxima: $P(\omega_1) = P(\omega_2) = \ldots = P(\omega_{n_c})$
- Impureza mínima (nodo puro): $P(\omega_i) = 1$





Criterios de medida de impureza

- Impureza de Entropía/Información (ID3, C4.5)

$$i(N) = -\sum_{j=1}^{c} P(w_j) \log_2 P(w_j)$$

$$i(N)_{\min} = 0 \qquad i(N)_{máx} = \log_2 c \qquad i(N) \geq 0$$

Criterios de medida de impureza

- Impureza de Gini (varianza)

$$i(N) = P(\omega_1)P(\omega_2)$$

← Dos clases

Criterios de medida de impureza

- Impureza de Gini (varianza)

$$i(N) = P(\omega_1)P(\omega_2)$$ ← Dos clases

$$i(N) = \sum_{i \neq j}^{c} P(\omega_i)P(\omega_j)$$ ← *c* clases

$$i_{\min}(N) = 0 \qquad i_{\min}(N) = \frac{c-1}{c}$$

Impureza de Gini - Interpretación

- Tasa de error esperado en el nodo N si la clase a asignar se sortea en forma aleatoria usando la distribución de clases presente en el nodo N.

$$P(w_1)(1-P(w_1)) = P(w_1)(P(w_2)+P(w_3)...+P(w_c))$$

$$\sum_i P(w_i)(1-P(w_i)) = \sum_i P(w_i) - \sum_i P(w_i)^2 = 1 - \sum_i P(w_i)^2$$

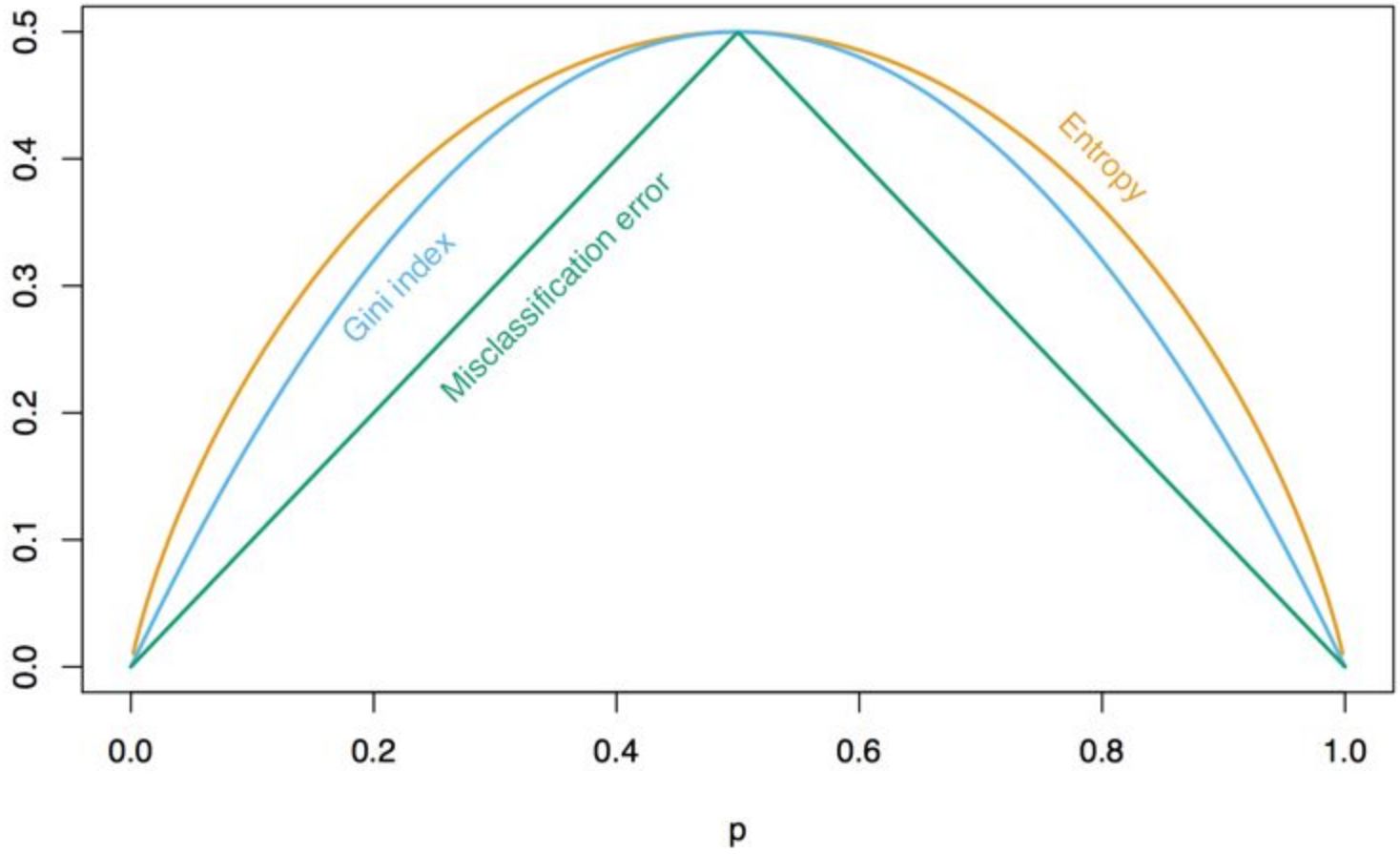
Criterios de medida de impureza

- Impureza de error de clasificación

$$i(N) = 1 - \max_j (P(w_j))$$

- Error de clasificación al asignar a la clase de la mayoría.
- Es la medida más "picuda" de las tres cuando las probabilidades son iguales.
- Derivadas discontinuas (puede complicar búsqueda de decisión óptima)

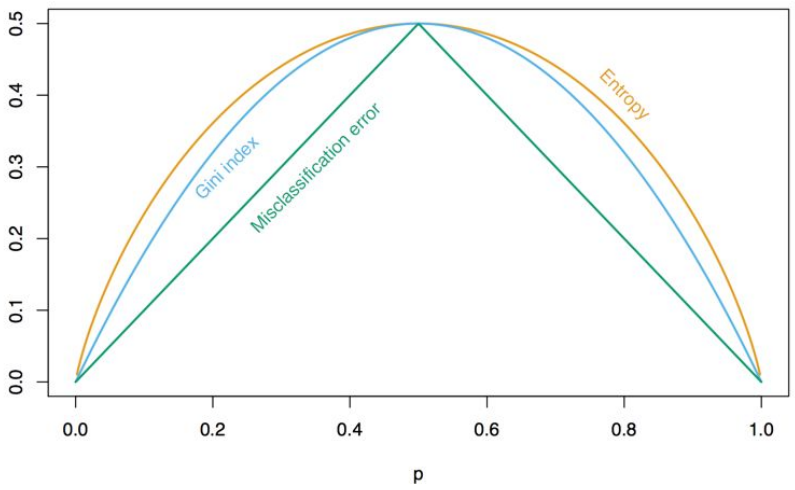
Criterios de medida de impureza



Impureza en un nodo para un problema binario de clasificación como función de la proporción de datos de una clase p.
(Valores renormalizados para visualización. Fig Hastie.)

Criterios de medida de impureza

- Entropía y Gini más sensibles
- Gini más "picudo" cuando las probabilidades son iguales.
- Entropía o Gini se usan para crecer el árbol y error de clasificación para podarlo.

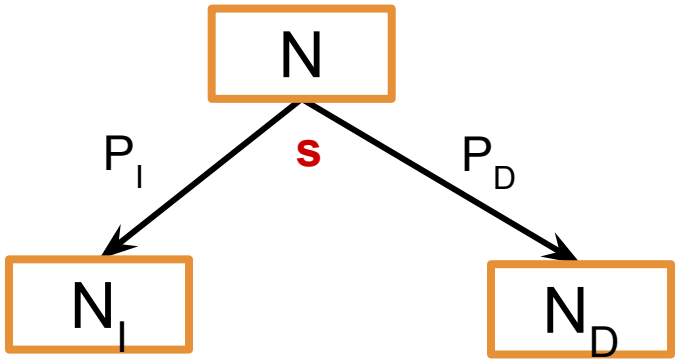


Criterios de medida de impureza

- Entropía y Gini más sensibles
- Gini más "picudo" cuando las probabilidades son iguales.
- Entropía o Gini se usan para crecer el árbol y error de clasificación para podarlo.
- Ej.:

	Error de clasificación	Gini	Entropía
(90,10)	0,1	0,09	0,33
(45,5) (5,45)	0,1	0,09	0,33
(70,0) (10,20)	0,1	0,066	0,19

Bondad de una partición



$$P_I = \frac{n_I}{n}$$

Bondad de la partición **s** en un nodo N

→ Decrecimiento en impureza

$$\Delta i(N) = i(N) - P_I i(N_I) - (1 - P_I) i(N_D)$$

Criterios de partición

- ¿Cuál es la mejor característica? La que maximiza el decremento de impureza:

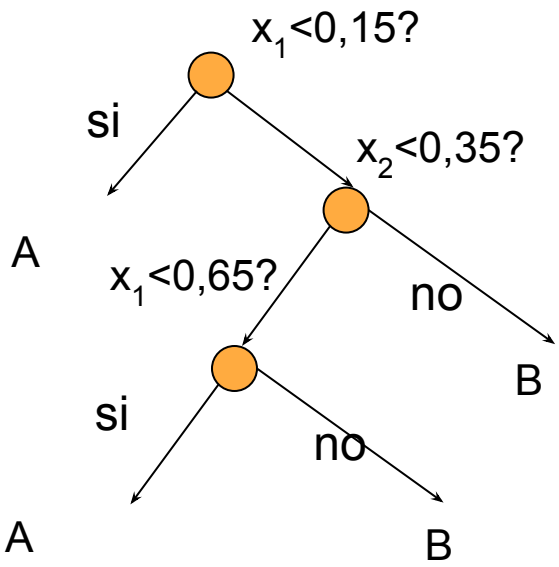
$$\Delta_i(N) = i(N) - P_I i(N_I) - (1 - P_I) i(N_D)$$

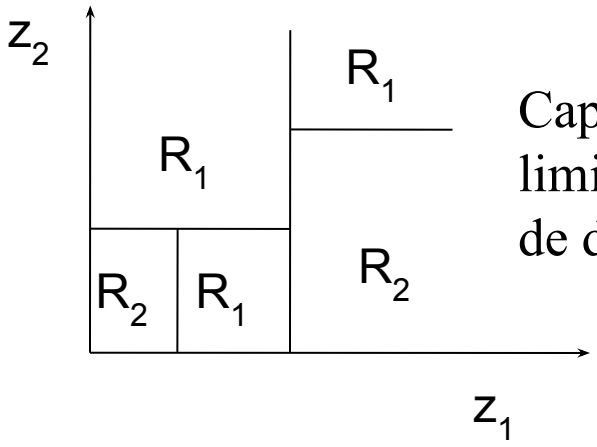
- Características nominales (color verde?) → probar con todas y seleccionar la de mayor Δ_i .
- Decisiones sobre características ordinales o continuas: primero encontrar el límite de decisión óptimo para cada característica y luego elegir la mejor.

Incorporación de atributos continuos

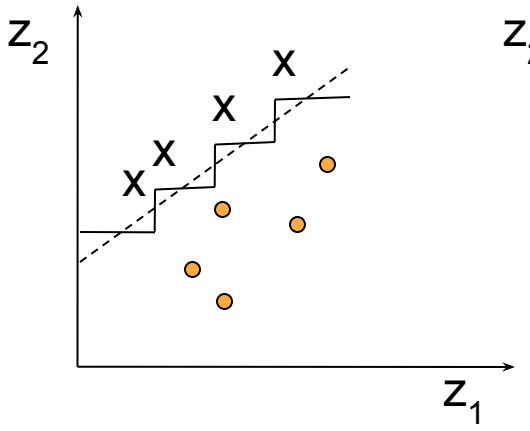
- Requiere la discretización de las características para poder incluirlas en el proceso de aprendizaje.
- Se genera una característica booleana comparando la característica continua contra un umbral. i.e., pregunta del tipo: ¿ $x_i < s$?
- Se busca el umbral s que produzca la mejor ganancia de entropía (impureza)

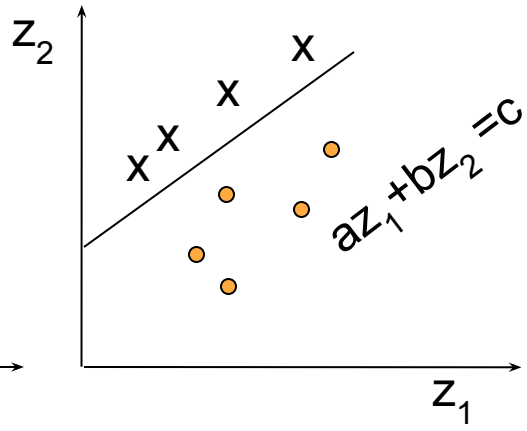
Límites de decisión





Capacidad expresiva limitada de los límites de decisión





Obs: Se puede volver a usar una característica.

Pierde facilidad de interpretación

Optimalidad de la solución

- **Optimalidad de decisión local**: Elegir el mejor atributo en cada nodo) **NO** garantiza la optimalidad global (por ejemplo el número mínimo de nodos)

Optimalidad de la solución

- **Optimalidad de decisión local**: Elegir el mejor atributo en cada nodo) **NO** garantiza la optimalidad global (por ejemplo el número mínimo de nodos)
- La elección de la función de impureza no es tan determinante como la elección del criterio de parada y los métodos de poda.
- Árbol muy grande sobreajustará (overfitting) a los datos, muy pequeño puede perder capacidad de discriminación.

Optimalidad de la solución

- **Optimalidad de decisión local**: Elegir el mejor atributo en cada nodo) **NO** garantiza la optimalidad global (por ejemplo el número mínimo de nodos)
- La elección de la función de impureza no es tan determinante como la elección del criterio de parada y los métodos de poda.
- Árbol muy grande sobreajustará (overfitting) a los datos, muy pequeño puede perder capacidad de discriminación.
- Tamaño del árbol: parámetro que gobierna la complejidad del modelo. Debe elegirse de acuerdo a los datos. Se busca árbol más simple, más compacto. Occam´s razor.

Criterios de Parada

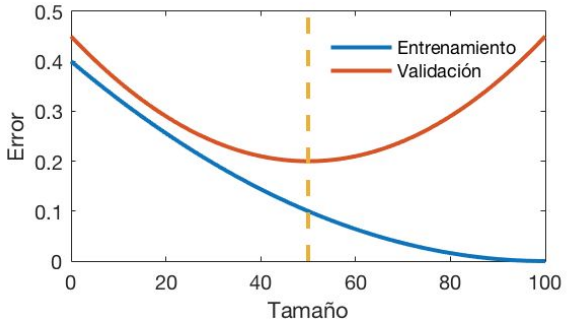
- Hacen crecer el árbol hasta el mínimo de impureza (suma de impurezas en cada hoja)
  - Desventaja: sobre-ajuste (overfiting)
- Alternativas:
  - Pre-podado: Detener el crecimiento
  - Pos-podado: Crecer hasta el límite y luego podar

Criterios de Parada

1. Validación cruzada: partir el set de datos en entrenamiento (90%), test/validación (10%).

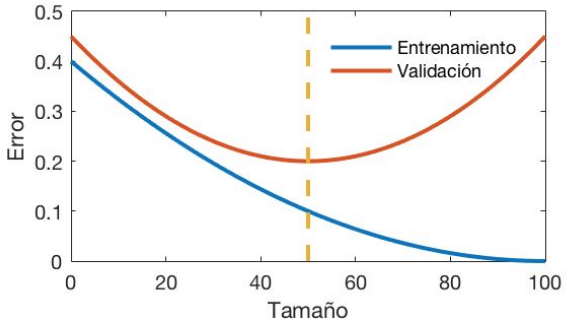
   Desventaja: Menos datos para entrenar.

   Se para de particionar cuando error comienza a crecer



Criterios de Parada

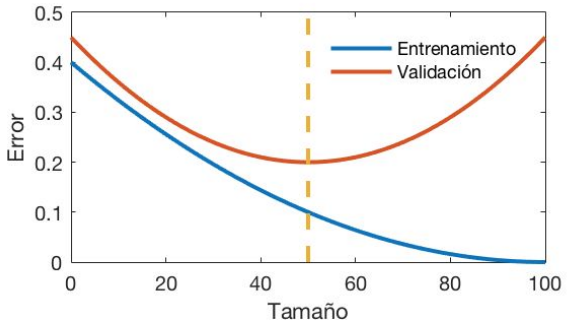
1. Validación cruzada: partir el set de datos en entrenamiento (90%), test/validación (10%).
   Desventaja: Menos datos para entrenar.
   Se para de particionar cuando error comienza a crecer



2. Umbral sobre la reducción de impureza: $\Delta_i(N) \leq \beta$. Árbol depende del β elegido por diseñador. (Se usa todo el conjunto de datos y pueden existir hojas en distintos niveles del árbol)

Criterios de Parada

1. Validación cruzada: partir el set de datos en entrenamiento (90%), test/validación (10%).
   Desventaja: Menos datos para entrenar.
   Se para de particionar cuando error comienza a crecer



2. Umbral sobre la reducción de impureza: $\Delta_i(N) \leq \beta$. Árbol depende del β elegido por diseñador. (Se usa todo el conjunto de datos y pueden existir hojas en distintos niveles del árbol)

3. Umbral sobre la cantidad de patrones en un nodo (*balanced*).
   Continuar creciendo el árbol con "pocos" patrones produciría sobre-ajuste. (Particiones chicas en zonas densas y grandes en zonas dispersas similar k-vecinos).

Criterios de Parada

4. **Penalizar la complejidad del árbol** (además del ajuste)

$$\min\left(C_\alpha(T)\right) = \min\left(\sum_{m=1}^{|T|} i(N) + \alpha|T|\right)$$

- Tamaño: # nodos, #hojas . Depende de elección de $\alpha>0$
- Minimiza la suma de la complejidad del modelo y la descripción (precisión) de los patrones dado el modelo.
- Dificultad determinar relación entre α y desempeño del clasificador .

Criterios de Parada

5. **Test de hipótesis.** Mide si reducción de impureza (Δi) es estadísticamente significativa (si difiere respecto a una partición aleatoria)

Criterios de Parada

5. **Test de hipótesis.** Mide si reducción de impureza (Δi) es estadísticamente significativa (si difiere respecto a una partición aleatoria)

➔ Supongamos hay ***n*** patrones (n_1 de w_1,n_2 de w_2) y que una determinada partición **s**, envía P***n*** a la izquierda y (1-P)***n*** a la derecha. Partición aleatoria debería enviar Pn_1 de w_1 *y* Pn_2 de w_2 a la izquierda (el resto a la derecha).

Criterios de Parada

5. **Test de hipótesis.** Mide si reducción de impureza (Δi) es estadísticamente significativa (si difiere respecto a una partición aleatoria)

- Supongamos hay $\boldsymbol{n}$ patrones (n_1 de w_1, n_2 de w_2) y que una determinada partición **s**, envía P$\boldsymbol{n}$ a la izquierda y (1-P)$\boldsymbol{n}$ a la derecha. Partición aleatoria debería enviar Pn_1 de w_1 y Pn_2 de w_2 a la izquierda (el resto a la derecha).
- Podemos medir la desviación respecto a la hipótesis nula mediante el estadístico de Pearson (chi-squared).
- Test χ^2: (o Pearson). Determinar si la distribución de eventos observados en una muestra es consistente con una cierta distribución teórica.

Criterios de Parada: Test de Hipótesis

$$C = 2 \qquad w_1, w_2 \qquad n = n_1 + n_2$$

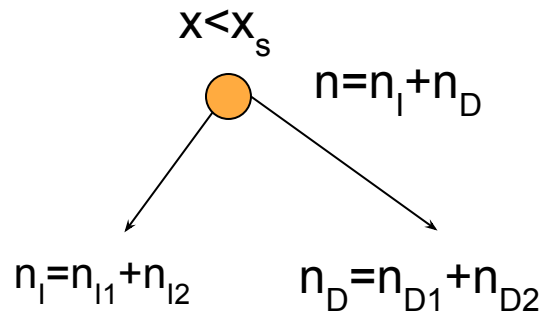
$$\chi_I^{\,2} = \frac{(n_{I1} - n_{e1})^2}{n_{e1}} + \frac{(n_{I2} - n_{e2})^2}{n_{e2}}$$

Valor Esperado si hay asignación aleatoria

$$n_{e1} = P n_1 = \frac{n_I}{n} n_1$$

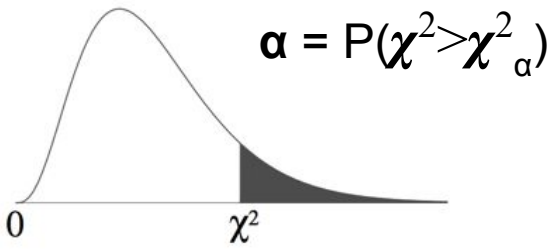
queremos ver si la partición s es distinta de randomica.

$$\chi^2 = \frac{\chi_I^{\,2} + \chi_D^{\,2}}{2} = \frac{n}{2n_D}\chi_I^{\,2} = \frac{n}{2n_I}\chi_D^{\,2}$$



- Hipótesis nula H_0: Distribuciones iguales (asignación aleatoria)
- Umbral para cierto nivel de confianza

$$\chi^2_{0.01(1)} = 6.64$$



Nivel de confianza: **α** Grados de libertad: c-1

$\chi^2 > 6.64$ Rechazamos H_0 : hacemos ramificación (si

$\chi^2 < 6.64$ Aceptamos H_0 : se detiene crecimiento (nodo terminal)

η_{χ^2} : Umbral establecido por el usuario

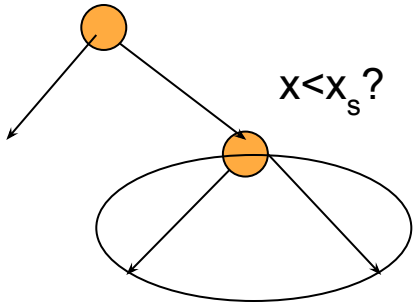
$\eta_{\chi^2} = 0$ Árbol completo

$\eta_{\chi^2} = 10$ "Podado" severo (pocos nodos)

$\chi^2_{.900}$	$\chi^2_{.100}$	$\chi^2_{.050}$	$\chi^2_{.025}$	$\chi^2_{.010}$	$\chi^2_{.005}$
0.016	2.706	3.841	5.024	6.635	7.879

Métodos de Poda (*pruning*)

- **Efecto horizonte**: al detener el crecimiento podemos perder ramificaciones posteriores beneficiosas.
- La poda permite que un subárbol de un nodo permanezca y la otra desaparezca, mientras que detener el crecimiento poda ambas ramas simultáneamente
- Alternativa: crecer el árbol completamente y **podarlo** luego. Preferible si es computacionalmente tolerable.
- **Idea**. Estimar error con y sin la ramificación y decidir si vale la pena la ramificación. Si no, unir los nodos.



Métodos de (pos-)poda

Remplazo de sub-árbol

- Conjunto de datos independiente para podado (partir el conjunto de datos disponibles en dos).
- Se busca eliminar el sobreajuste (overfitting)

Métodos de (pos-)poda

Remplazo de sub-árbol

- Conjunto de datos independiente para podado (partir el conjunto de datos disponibles en dos).
- Se busca eliminar el sobreajuste (overfitting)
- Comenzando desde las hojas y hacia la raíz:
  - Se **remplaza** un nodo con hoja etiquetada por clase mayoritaria.
  - Se calcula el error en el conjunto de podado si es menor se remplaza el nodo por hoja.
- Se obtiene un sub-árbol óptimo para conjunto de poda.

Poda por mínimo costo-complejidad

- *Complejidad* de un sub-árbol: Número de nodos terminales (hojas), $|T|$.
- Error de clasificación $R(T)$.
- Medida de costo-complejidad:

$$R_\alpha(T) = R(T) + \alpha |T|$$

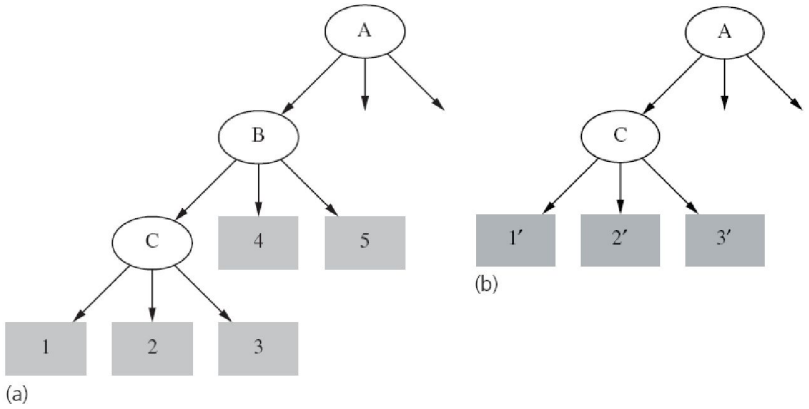
- $\alpha \geq 0$: parámetro de complejidad

$$R_\alpha(T(\alpha)) = \min_{T \leq T_{\max}} R_\alpha(T)$$

Métodos de Poda

Elevación de Sub-árbol

- Usando todo los datos para entrenamiento
- Es más compleja y no necesariamente siempre es útil: Usada en C4.5.
- Estima el error cometido en un nodo cuando sustituyo sub-árbol por una de sus ramas. Top/down.

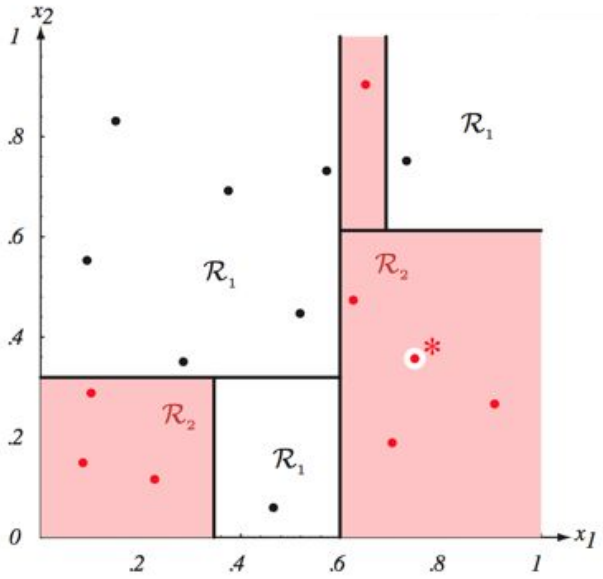


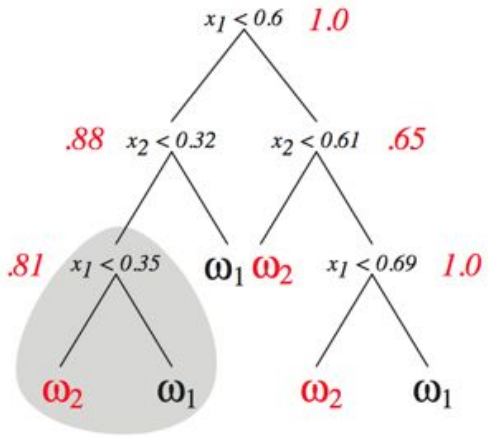
Árboles de Decisión

- **Inestabilidad**: sensibles al conjunto de entrenamiento, alta varianza. Pequeño cambio en los datos puede generar gran cambio en particiones, haciendo interpretación precaria (ejemplo a continuación)
- **Alternativas**: Bagging, Random Forest
- Priors y costos: Es posible incluir priors o costos pesando los patrones de entrenamiento con el prior o los costos.

  Ej: Índice de Gini: $i(N) = \sum_{i,j} \lambda_{i,j} P(w_i) P(w_j)$

EJEMPLO - Clasificación binaria - 8 muestras



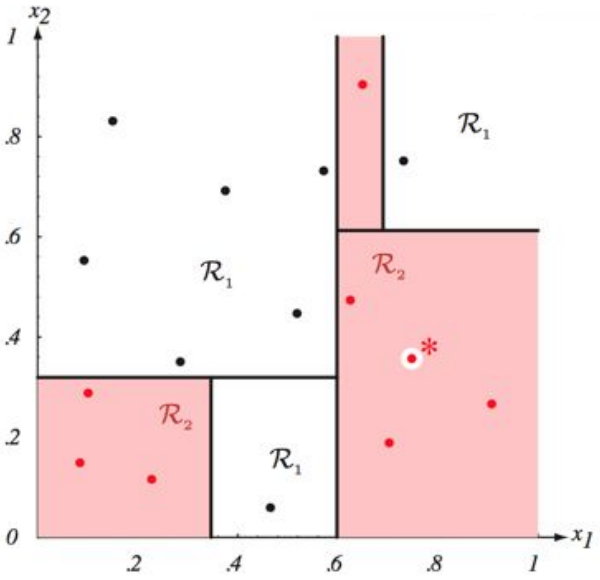


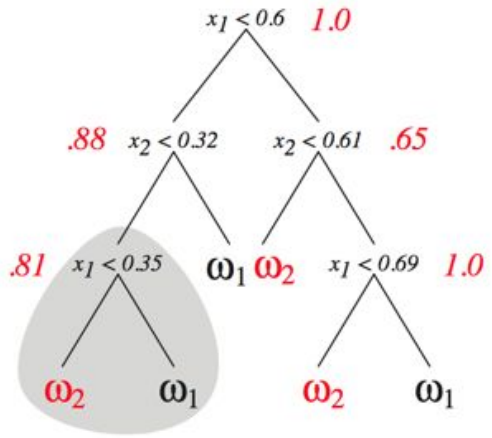
ω_1 (black)	
x_1	x_2
.15	.83
.09	.55
.29	.35
.38	.70
.52	.48
.57	.73
.73	.75
.47	.06

ω_2 (red)	
x_1	x_2
.10	.29
.08	.15
.23	.16
.70	.19
.62	.47
.91	.27
.65	.90
.75	.36*

Impureza de Entropía

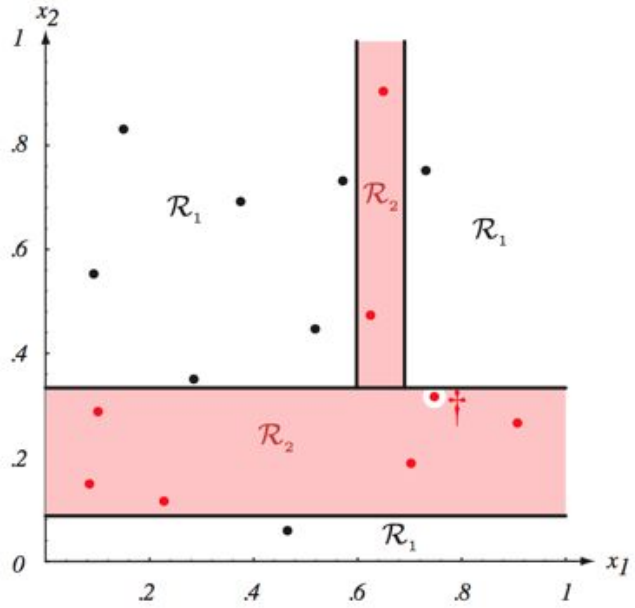
EJEMPLO - Clasificación binaria - 8 muestras

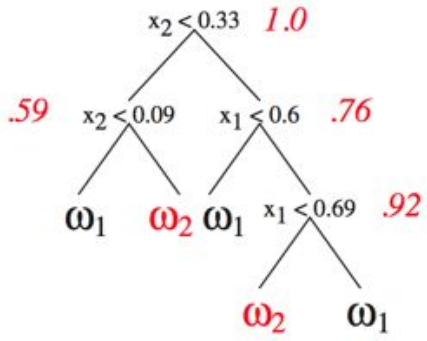




ω_1 (black)		ω_2 (red)	
x_1	x_2	x_1	x_2
.15	.83	.10	.29
.09	.55	.08	.15
.29	.35	.23	.16
.38	.70	.70	.19
.52	.48	.62	.47
.57	.73	.91	.27
.73	.75	.65	.90
.47	.06	.75	.36* (.32†)

Impureza de Entropía





Inestable!

Datos faltantes (en entrenamiento)

- Construir el árbol usando los patrones que tienen definida la característica (puede ser bastante restrictivo)

Datos faltantes (en entrenamiento)

- Construir el árbol usando los patrones que tienen definida la característica (puede ser bastante restrictivo)
- Si en un nodo N dado hay un patrón $\mathbf{x}=\{x_1, x_2, x_3\}$ con una característica x_2 faltante, se puede estimar *i(N)* con n patrones para x_1 y x_3 y con (n-1) para x_2.
- Se utiliza la característica que decrece i(N) en mayor cantidad

Datos faltantes (en *testing*)

- Supongamos que queremos construir un árbol capaz de procesar (en testing) muestras con datos faltantes

Datos faltantes (en *testing*)

- Supongamos que queremos construir un árbol capaz de procesar (en testing) muestras con datos faltantes
- En un nodo dado N, luego de elegir la mejor característica para una ramificación (característica *primaria*), se eligen características suplentes en orden, considerando la correlación entre características.

Datos faltantes (en *testing*)

- Supongamos que queremos construir un árbol capaz de procesar (en testing) muestras con datos faltantes
- En un nodo dado N, luego de elegir la mejor característica para una ramificación (característica *primaria*), se eligen características suplentes en orden, considerando la correlación entre características.

Correlación = # Patrones a la izquierda por ambas + #Patrones a la derecha por ambas

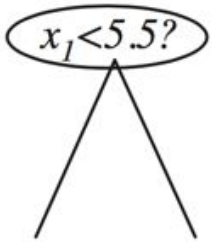
- El objetivo es tratar de replicar la división dada por la característica *primaria*
- En la clasificación si el atributo sobre el que hay que decidir falta se usa el siguiente suplente.

$$\omega_1: \quad \mathbf{x}_1 \begin{pmatrix} 0 \\ 7 \\ 8 \end{pmatrix}, \quad \mathbf{x}_2 \begin{pmatrix} 1 \\ 8 \\ 9 \end{pmatrix}, \quad \mathbf{x}_3 \begin{pmatrix} 2 \\ 9 \\ 0 \end{pmatrix}, \quad \mathbf{x}_4 \begin{pmatrix} 4 \\ 1 \\ 1 \end{pmatrix}, \quad \mathbf{x}_5 \begin{pmatrix} 5 \\ 2 \\ 2 \end{pmatrix}$$

$$\omega_2: \quad \mathbf{y}_1 \begin{pmatrix} 3 \\ 3 \\ 3 \end{pmatrix}, \quad \mathbf{y}_2 \begin{pmatrix} 6 \\ 0 \\ 4 \end{pmatrix}, \quad \mathbf{y}_3 \begin{pmatrix} 7 \\ 4 \\ 5 \end{pmatrix}, \quad \mathbf{y}_4 \begin{pmatrix} 8 \\ 5 \\ 6 \end{pmatrix}, \quad \mathbf{y}_5 \begin{pmatrix} 9 \\ 6 \\ 7 \end{pmatrix}$$

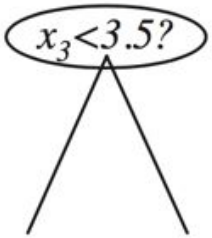
Duda-Hart

primary split



$\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{y}_1$ $\quad \boldsymbol{y}_2, \boldsymbol{y}_3, \boldsymbol{y}_4, \boldsymbol{y}_5$

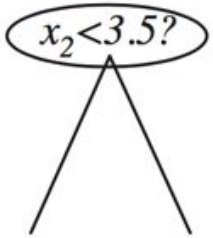
first surrogate split



$\boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{y}_1$ $\quad \boldsymbol{y}_2, \boldsymbol{y}_3, \boldsymbol{y}_4, \boldsymbol{y}_5,$ $\boldsymbol{x}_1, \boldsymbol{x}_2$

predictive association with primary split = 8

second surrogate split



$\boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{y}_1,$ $\quad \boldsymbol{y}_3, \boldsymbol{y}_4, \boldsymbol{y}_5,$ $\boldsymbol{y}_2$ $\quad \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$

predictive association with primary split = 6

¿Qué clasificador es mejor?

Algunos criterios:

- Criterio de crecimiento: impureza de entropía funciona bien
  - regla defecto.
- Podado preferible a parada con validación cruzada. Aunque podado de conjunto de datos grandes puede ser inabordable.
- No hay un árbol superior a otro...
- Árboles de decisión desempeño "similar" a otros métodos como redes neuronales, k-vecinos.
- Particularmente útiles con datos no métricos.

Fortalezas y debilidades de Árboles

Interpretabilidad de las reglas de decisión

Bajo costo computacional para clasificar

Pueden usar características continuas y categóricas

Entrenamiento computacionalmente caro (post-poda)

No tratan bien regiones no-rectangulares.