Árboles y Reglas de Decisión

Aprendizaje Automático

Ingeniería Informática

Fernando Fernández Rebollo y Daniel Borrajo Millán

Grupo de Planificación y Aprendizaje (PLG)
Departamento de Informática
Escuela Politécnica Superior
Universidad Carlos III de Madrid

27 de febrero de 2009

En Esta Sección:

Aprendizaje de árboles de decisión. ID3

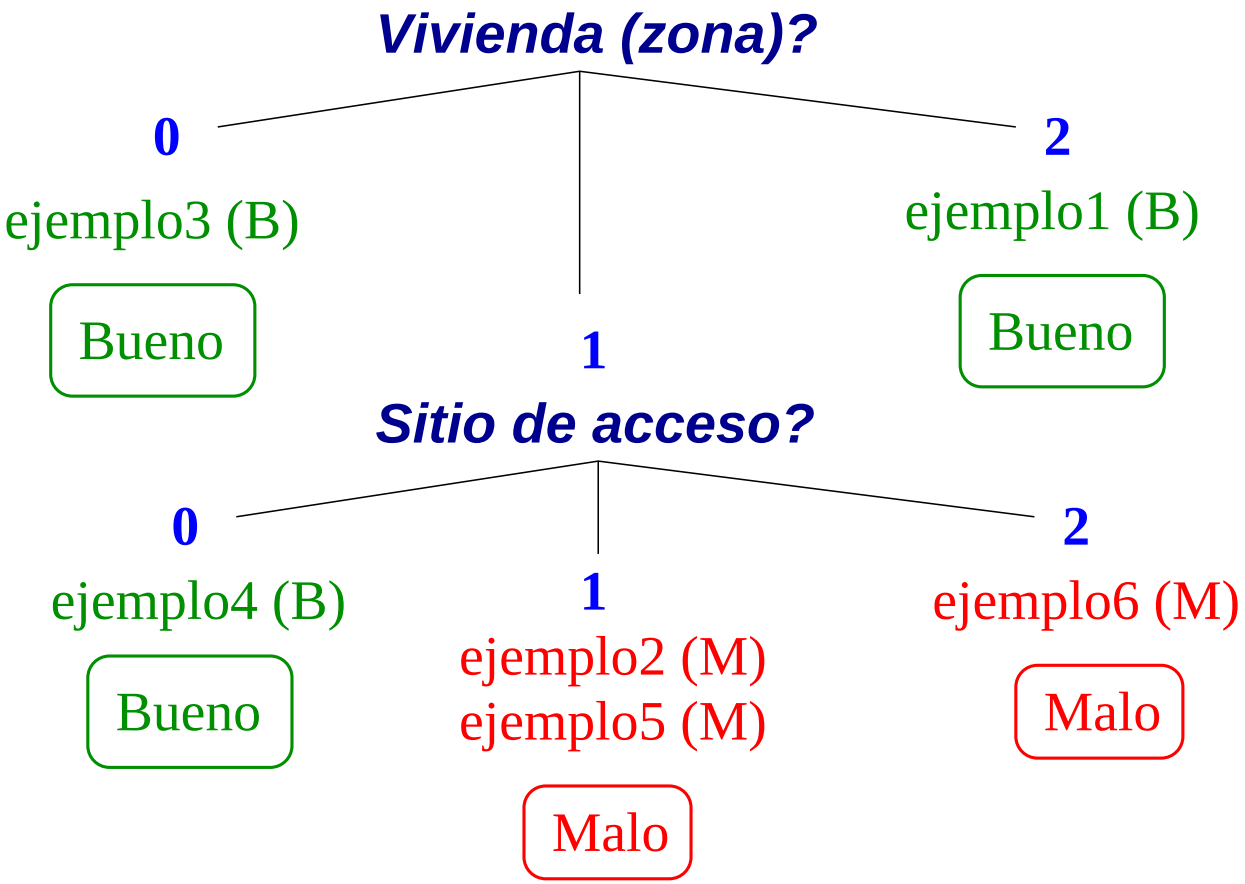
(Quinlan, 83)

- CLS (Hunt, Marin, y Stone, 66) fue el precursor de ID3
  - Utilizaba sólo atributos binarios
  - Tenía heurísticas para decidir qué atributo escoger
- Conjunto de técnicas que han tenido mucho éxito comercial
- Genera árboles de decisión a partir de ejemplos de partida
- Intenta encontrar el árbol más sencillo que separa mejor los ejemplos
- Utiliza la entropía para elegir

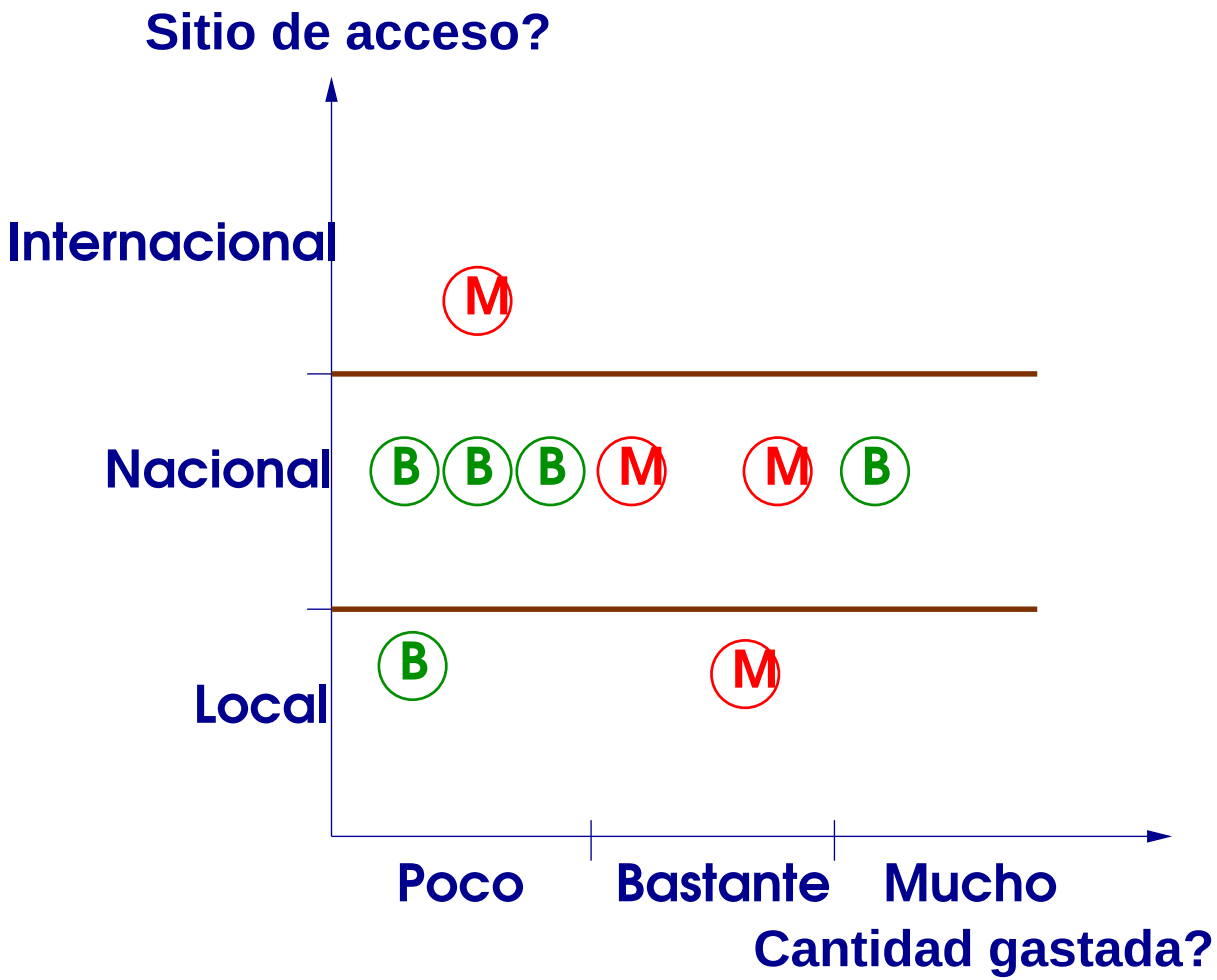
Ejemplo de entrada

Ejemplo	Sitio de acceso A_1	1ª cantidad gastada A_2	Vivienda (zona) A_3	Última compra A_4	Clase
1	1	0	2	Libro	Bueno
2	1	0	1	Disco	Malo
3	1	2	0	Libro	Bueno
4	0	2	1	Libro	Bueno
5	1	1	1	Libro	Malo
6	2	2	1	Libro	Malo

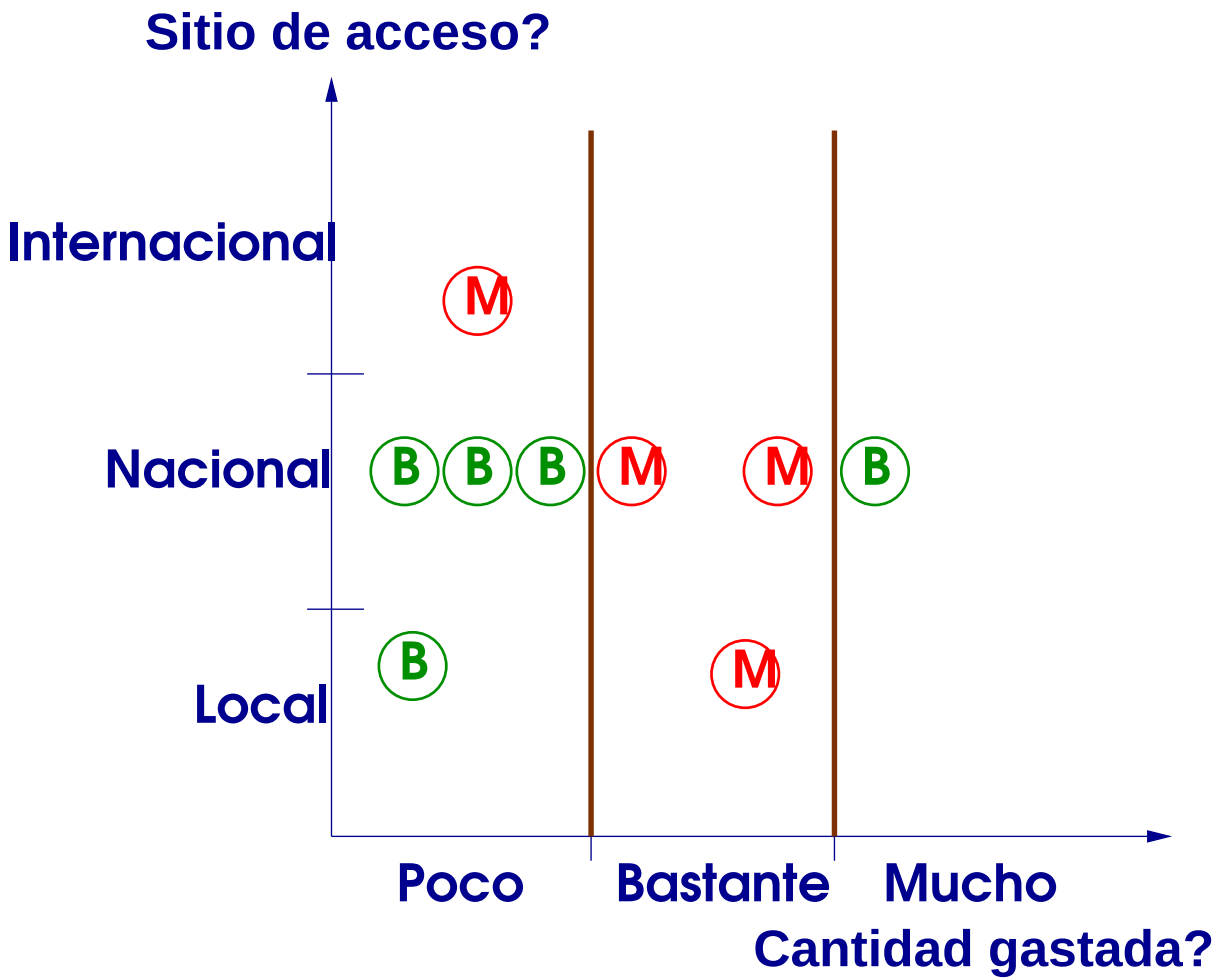
Ejemplo de árbol de decisión



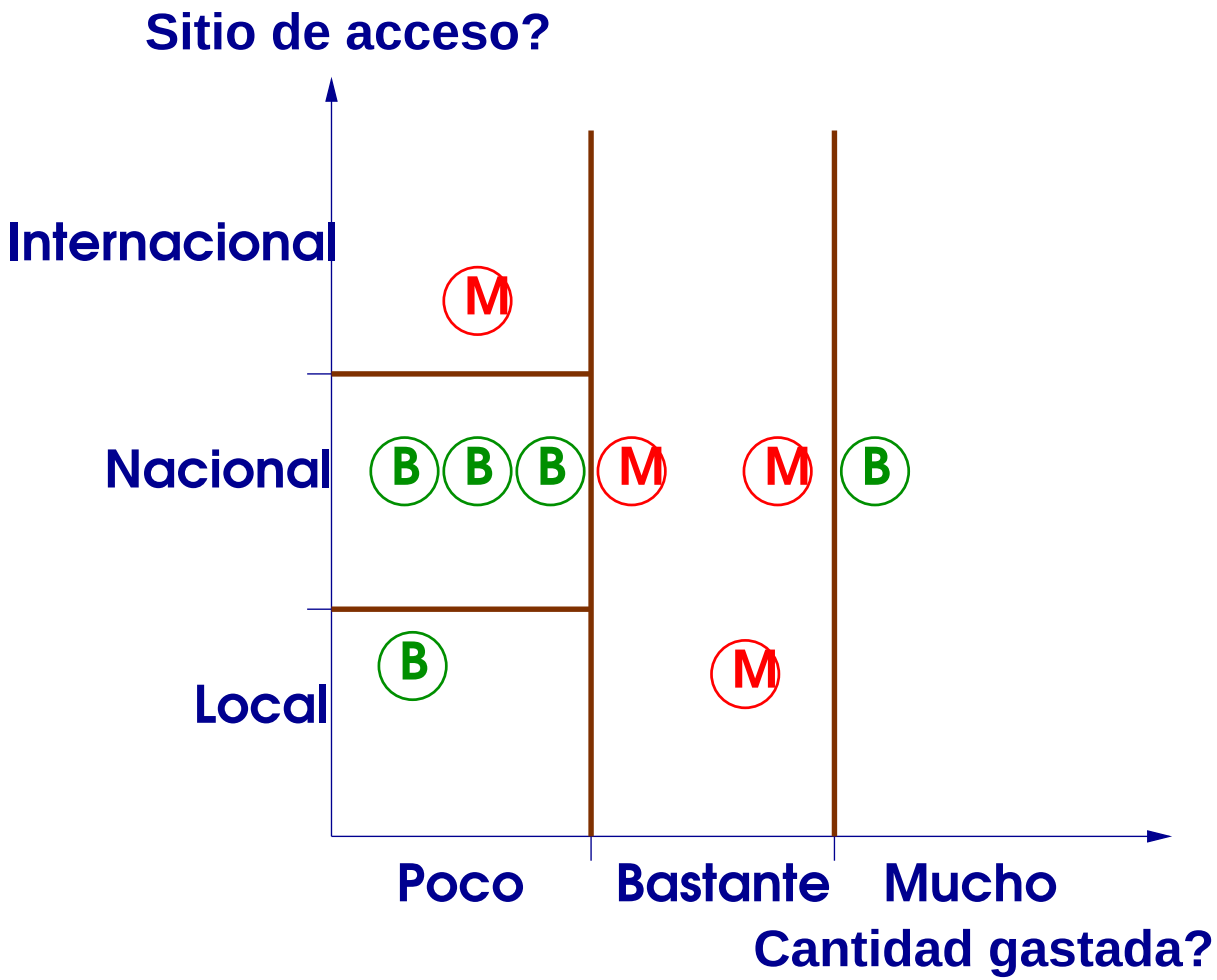
Estrategia del ID3



Estrategia del ID3



Estrategia del ID3



Algoritmo

1. Seleccionar el atributo A_i que maximice la ganancia $G(A_i)$
2. Crear un nodo para ese atributo con tantos sucesores como valores tenga
3. Introducir los ejemplos en los sucesores según el valor que tenga el atributo A_i
4. Por cada sucesor,

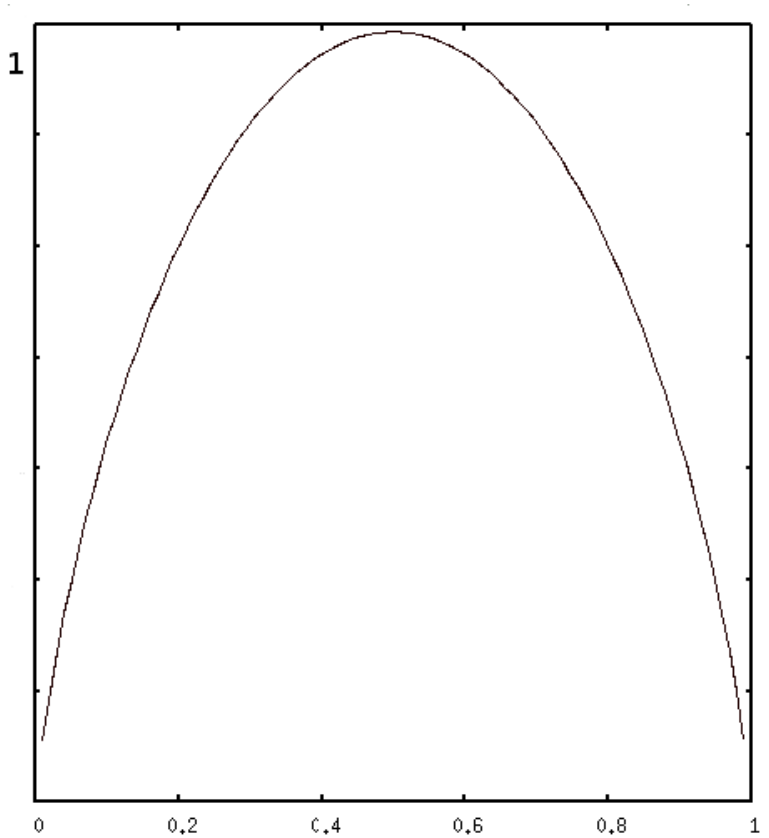
   Si sólo hay ejemplos de una clase c_k
   Entonces etiquetarlo con c_k
   Si no, llamar al ID3 con una tabla formada por los ejemplos de ese nodo, eliminando la columna del atributo A_i

Heurística

- Seleccionar el atributo que mejor separe (ordene) los ejemplos de acuerdo a las clases
- La entropía es una medida de cómo está ordenado el universo
- La teoría de la información (basada en la entropía) calcula el número de bits (información, preguntas sobre atributos) que hace falta suministrar para conocer la clase a la que pertenece un ejemplo
- Entropía de clasificación de una colección de datos que pertenecen a una de entre dos categorías (clasificación binaria): $I \sim -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$
  donde $p_{\oplus}$ es la proporción de ejemplos positivos sobre el total, y $p_{\ominus}$ es la proporción de ejemplos negativos sobre el total, con $p_{\oplus} + p_{\ominus} = 1$

Entropía

- Si $p_{\oplus} = p_{\ominus} = 0{,}5$, entonces la entropía es máxima
- La entropía tiende a 0 cuanto más se diferencian las probabilidades "a priori"
- Con múltiples clases: $I \sim \sum_{i=1}^{c} -p_i \log_2 p_i$
- Objetivo: minimizar la entropía



Ganancia de información de un atributo

- Esperanza de reducción de entropía cuando se divide el conjunto de datos original según el atributo dado

$$A = \arg\max_{a \in \mathcal{A}} G(a) = \arg\max_{a \in \mathcal{A}} [I - I(a)] = \arg\min_{a \in \mathcal{A}} I(a)$$

  - Entropía del atributo A_i:

$$I(A_i) = \sum_{j=1}^{nv(A_i)} \frac{n_{ij}}{n} I_{ij}$$

  - Entropía de la partición j del atributo A_i:

$$I_{ij} = -\sum_{k=1}^{nc} \frac{n_{ijk}}{n_{ij}} \log_2 \frac{n_{ijk}}{n_{ij}}$$

Ejemplo de ID3

Ejemplo	Sitio de acceso A_1	1ª cantidad gastada A_2	Vivienda (zona) A_3	Última compra A_4	Clase
1	1	0	2	Libro	Bueno
2	1	0	1	Disco	Malo
3	1	2	0	Libro	Bueno
4	0	2	1	Libro	Bueno
5	1	1	1	Libro	Malo
6	2	2	1	Libro	Malo

Ejemplo de ID3

$$I(A_1) = \sum_{j=1}^{nv(A_1)} \frac{n_{ij}}{n} I_{ij} = \sum_{j=1}^{3} \frac{n_{ij}}{6} I_{ij} =$$
$$\frac{n_{10}}{6} I_{10} + \frac{n_{11}}{6} I_{11} + \frac{n_{12}}{6} I_{12} = \frac{1}{6} I_{10} + \frac{4}{6} I_{11} + \frac{1}{6} I_{12}$$
$$I_{10} = -\sum_{k=1}^{2} \frac{n_{10k}}{n_{10}} \log_2 \frac{n_{10k}}{n_{10}} = -\frac{1}{1} \log_2 \frac{1}{1} - \frac{0}{1} \log_2 \frac{0}{1} = 0$$
$$I_{11} = -\sum_{k=1}^{2} \frac{n_{11k}}{n_{11}} \log_2 \frac{n_{11k}}{n_{11}} = -\frac{2}{4} \log_2 \frac{2}{4} - \frac{2}{4} \log_2 \frac{2}{4} = 1$$
$$I_{12} = -\sum_{k=1}^{2} \frac{n_{12k}}{n_{12}} \log_2 \frac{n_{12k}}{n_{12}} = -\frac{0}{1} \log_2 \frac{0}{1} - \frac{1}{1} \log_2 \frac{1}{1} = 0$$

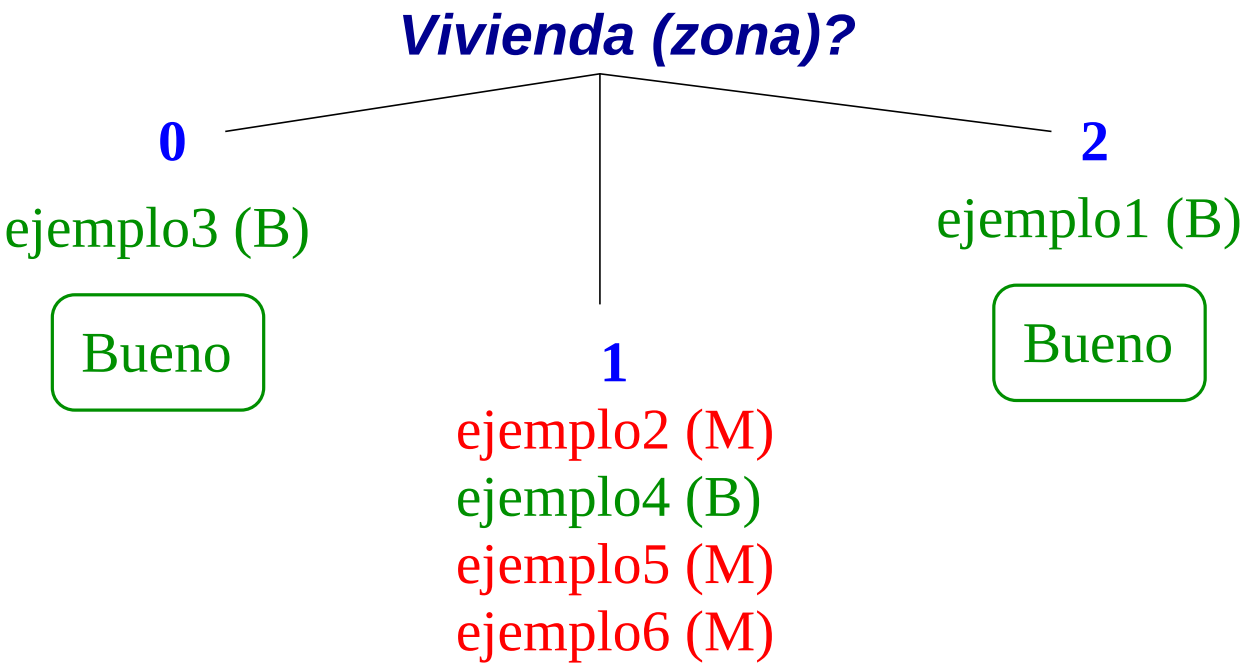
$$I(A_1) = \frac{1}{6} I_{10} + \frac{4}{6} I_{11} + \frac{1}{6} I_{12} = \frac{1}{6} 0 + \frac{4}{6} 1 + \frac{1}{6} 0 = 0,66$$

$$I(A_2) = \frac{2}{6} I_{20} + \frac{1}{6} I_{21} + \frac{3}{6} I_{22} = \frac{2}{6} 1 + \frac{1}{6} 0 + \frac{3}{6} (-\frac{2}{3} \log_2 \frac{2}{3} - \frac{1}{3} \log_2 \frac{1}{3}) = 0,79$$

$$I(A_3) = \frac{1}{6} I_{30} + \frac{4}{6} I_{31} + \frac{1}{6} I_{32} = \frac{1}{6} 0 + \frac{4}{6} (-\frac{1}{4} \log_2 \frac{1}{4} - \frac{3}{4} \log_2 \frac{3}{4}) + \frac{1}{6} 0 = 0,54$$

$$I(A_4) = \frac{1}{6} I_{4Disco} + \frac{5}{6} I_{4Libro} = \frac{1}{6} 0 + \frac{5}{6} (-\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5}) = 0,81$$

Ejemplo de ID3



Ejemplo de ID3

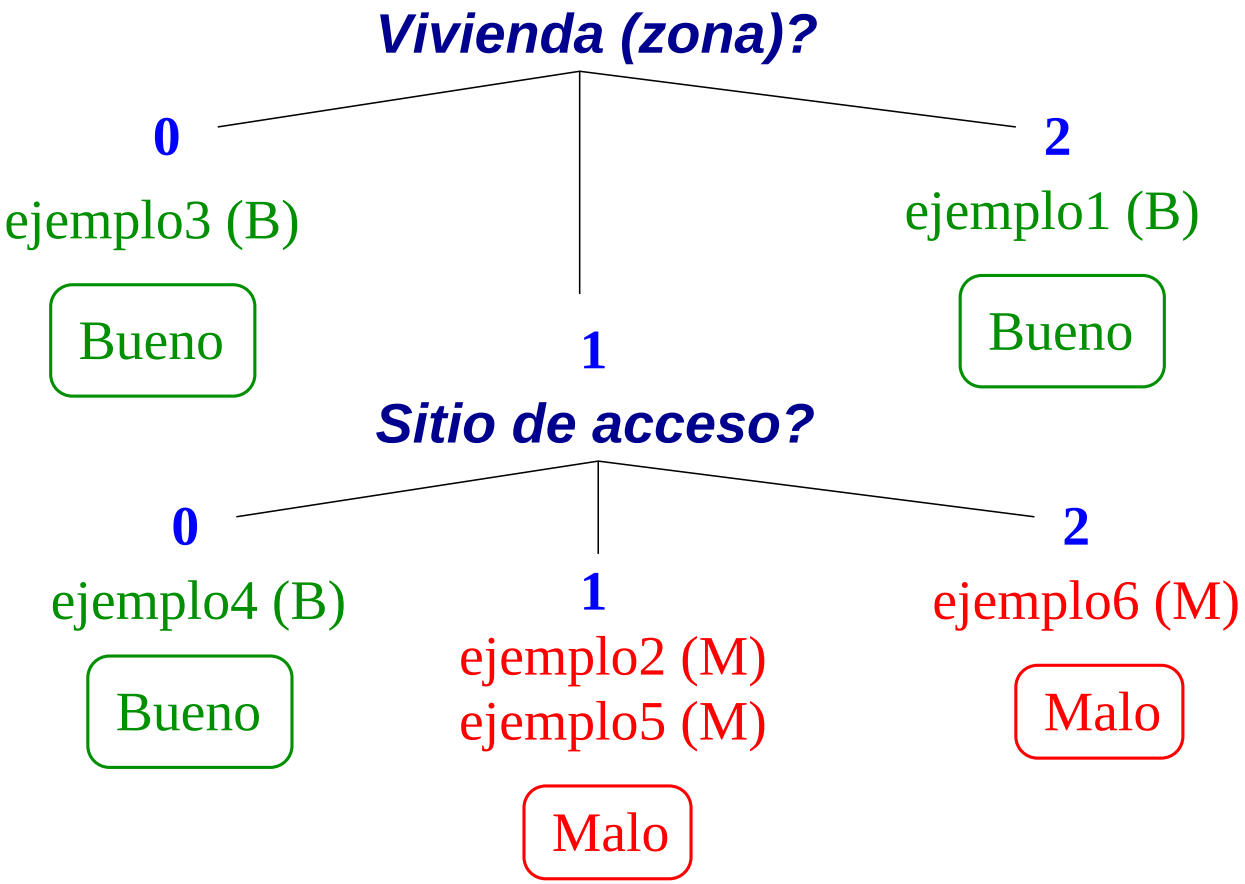
Ejemplo	Sitio de acceso A_1	1ª cantidad gastada A_2	Última compra A_4	Clase
2	1	0	Disco	Malo
4	0	2	Libro	Bueno
5	1	1	Libro	Malo
6	2	2	Libro	Malo

$$I(A_1) = \frac{1}{4} I_{10} + \frac{2}{4} I_{11} + \frac{1}{4} I_{12} = \frac{1}{4} 0 + \frac{2}{4} 0 + \frac{1}{4} 0 = 0$$

$$I(A_2) = \frac{1}{4} I_{20} + \frac{1}{4} I_{21} + \frac{2}{4} I_{22} = \frac{1}{4} 0 + \frac{1}{4} 0 + \frac{2}{4} 1 = 0,5$$

$$I(A_4) = \frac{1}{4} I_{4Disco} + \frac{3}{4} I_{4Libro} = \frac{1}{4} 0 + \frac{3}{4} (-\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3}) = 0,23$$

Ejemplo de ID3



Traducción a reglas

- Cualquier árbol de decisión se puede convertir a reglas
- Regla: estructura del tipo Si-Entonces
- Ejemplo
  SI Vivienda (zona)=1 Y Sitio de acceso=0
  ENTONCES Bueno
- Algoritmo: por cada rama del árbol, las preguntas y sus valores estarán en la parte izquierda de las reglas y la etiqueta del nodo hoja correspondiente será la parte derecha (clasificación)

Ejemplo de traducción a reglas

Si Vivienda (zona)=0
Entonces Bueno
Si Vivienda (zona)=1 y Sitio de acceso=0
Entonces Bueno
Si Vivienda (zona)=1 y Sitio de acceso=1
Entonces Malo
Si Vivienda (zona)=1 y Sitio de acceso=2
Entonces Malo
Si Vivienda (zona)=2
Entonces Bueno

ID3 como un problema de búsqueda

- Conjunto de estados: cada estado es un árbol de decisión
- Conjunto de operadores: el único operador es "introducir en un nodo la pregunta del atributo correspondiente"
- Estado inicial: árbol de decisión vacío
- Meta: árbol de decisión que separa los ejemplos de entrenamiento dependiendo de su clase
- Heurística: elegir aquel atributo que minimice la entropía

Algunas consideraciones sobre la búsqueda

- El espacio de búsqueda es completo: mantiene una hipótesis única, disyunción de conjunciones
- El algoritmo de búsqueda es incompleto: no realiza retroceso
  - pero, no se alcanzan mínimos locales si no se tiene ruido y se tienen todos los atributos relevantes
- Heurística basada en la estadística
  - es robusta al ruido: si un ejemplo es incorrecto, la estadística suavizará el efecto

Bias inductiva

- *Bias* en ID3
  - Preferir árboles con atributo con mayor información más cerca de la raíz, y árboles más cortos
  - *La navaja de Occam*: preferir siempre las hipótesis más sencillas que describan los datos
  - ID3 favorece atributos con muchos valores

	ID3	espacio de versiones
búsqueda	incompleta	completa
espacio de hipótesis	completo	incompleto
bias debida al	método de búsqueda	espacio de búsqueda
tipo de *bias*	preferencia	restrictiva

- Hay otros sistemas que combinan los dos tipos de *bias*. Por ejemplo, redes de neuronas

Dificultades del ID3

- ¿Cuándo se debe parar de subdividir el árbol? *sobreadecuación (overfitting), poda*
- ¿Qué se hace con valores contínuos de atributos? *peso*
- ¿Qué se hace con valores discretos con muchos valores? *día del cumpleaños*
- ¿Qué pasa si el coste de conocer el valor de un atributo no es constante? *presión sanguínea vs. biopsia*
- ¿Qué se hace cuando los ejemplos vienen incrementalmente? ID4 *e* ID5

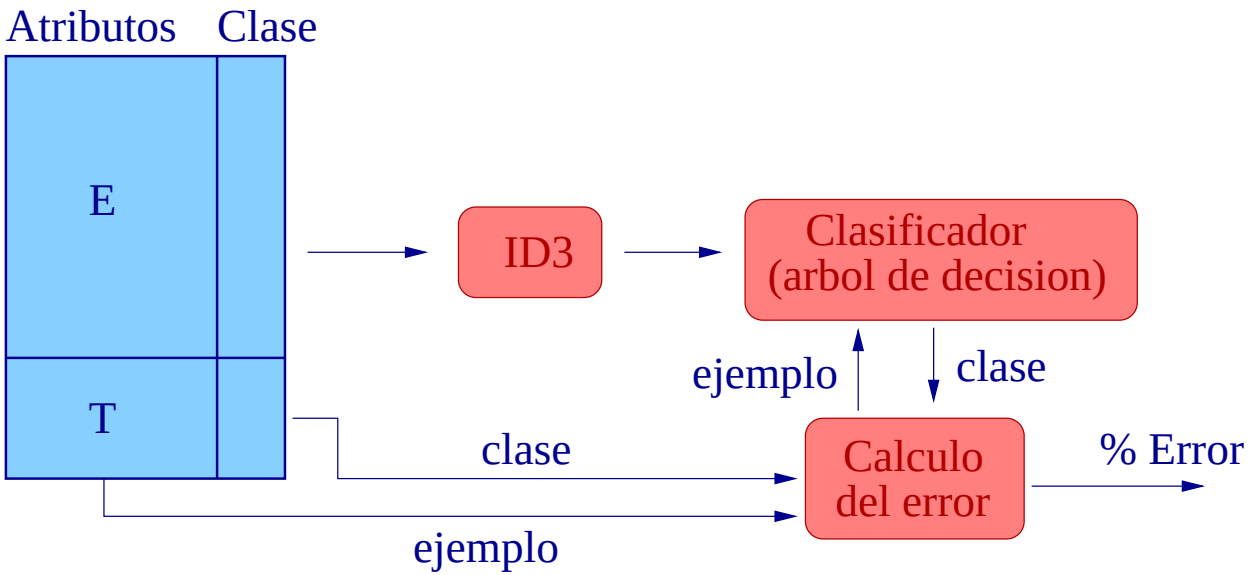
Dificultades (II)

- ¿Qué se hace cuando los ejemplos están representados en lógica de primer orden? *ILP*
- ¿Qué ocurre cuando hay atributos con valores desconocidos?
  - asignar el valor más probable
  - asignar distribución de probabilidad
- ¿Qué ocurre cuando las clases son contínuas? M5
- ¿Qué se hace cuando dos partes del árbol son iguales? *replicación*

Evaluación

- el conjunto de ejemplos se divide en dos partes: entrenamiento (E) y *test* (T)
- se aplica la técnica (p.e. el ID3) al conjunto de entrenamiento, generando un clasificador
- se calcula el número de errores (o aciertos) que el clasificador comete en el conjunto de *test*

Cálculo del error



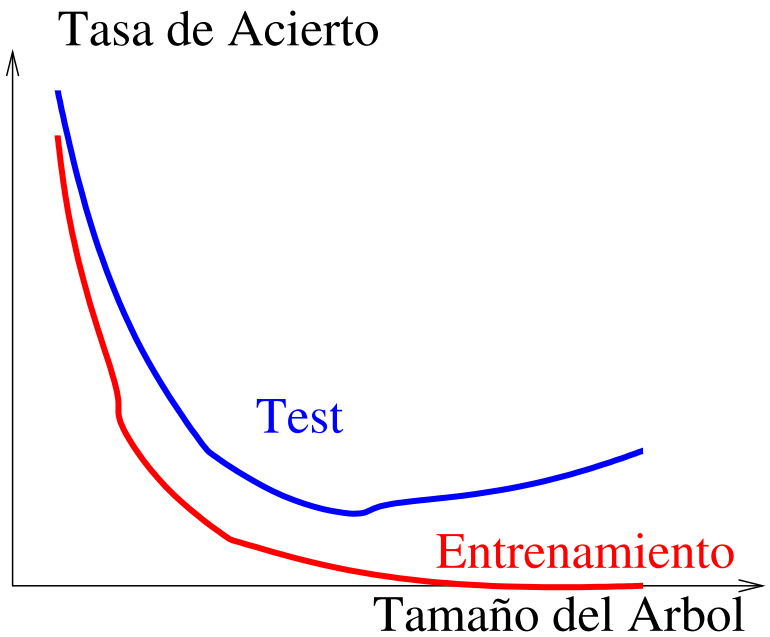
Validación cruzada

- Problema de la evaluación: sesgo de los conjuntos E y T seleccionados
- Solución: validación cruzada k-veces (*k-fold cross validation*)
  - Se divide el conjunto de ejemplos en k partes iguales, E_i
  - Se realiza lo siguiente k veces:
    - se entrena con $E - E_i$ (i=1..k)
    - se calcula el error con el E_i, e_i
  - Se estima la tasa de error haciendo la media de los errores

$$r = \sum_{i=1}^{k} \frac{e_i}{k}$$

Sobreadecuación (*Overfitting*)

Las hipótesis se refinan tanto que describen muy bien las instancias de aprendizaje, pero el error de clasificación crece en ejemplos externos



- Provocado por: ruido en los ejemplos, pocos ejemplos, o error en la representación
- Solución: poda

Pre-poda

- *Solución 1*: utilizar el χ^2 (Quinlan, 86)
  - no se divide un nodo si se tiene poca confianza en él (no es significativa la diferencia de clases)
  - ejemplo: cuando se tienen 40 ejemplos positivos y uno negativo
  - es muy conservador y puede hacer que se pare el árbol antes de lo que conviene
- *Solución 2*: generar las curvas y parar cuando la curva del conjunto de test empieza a subir

Post-poda

- *Solución 3*: generar el árbol y analizar recursivamente y desde las hojas qué preguntas se pueden eliminar sin que afecten al error de clasificación con el conjunto de *test*
- *Solución 4*: generar las reglas equivalentes y eliminar condiciones/reglas si el error de clasificación es menor sin ellas (C4.5)

Poda de reglas. C4.5 (Quinlan)

Convertir el árbol de decisión a un conjunto de reglas $\mathcal{R}$
error = error de clasificación con $\mathcal{R}$
Para cada regla $R_i \in \mathcal{R}$
    Para cada precondición p_j de $p(R_i)$
        nuevo-error = error al eliminar p_j de $p(R_i)$
        Si nuevo-error $\leq$ error
        Entonces error = nuevo-error
                eliminar p_j de $p(R_i)$
    Si $p(R_i)$ está vacío
    Entonces eliminar R_i

Atributos con valores continuos

- Se ordenan los valores del atributo, y se especifica la clase a la que pertenecen

peso	30	36	44	60	72	78
dodó	no	no	sí	sí	sí	no

- Hay dos puntos de corte, (36-44) y (72-78), en los que se pueden calcular los valores medio: 40 y 75
- Para crear el nodo de decisión
  - Se pueden crear atributos dinámicamente

$$peso < 40$$

  - Se puede hacer la distinción en el mismo nodo

$$peso < 40, 40 < peso < 75, 75 < peso$$

Atributos con muchos valores

- Por cada valor v del atributo A, se puede crear un atributo binario $(A = v)$
- Para mejorar en eficiencia, sólo se crean cuando son necesarios
- Dado que ID3 prefiere atributos con mayor número de valores, se les puede desfavorecer utilizando la medida Razón de ganancia (*GainRatio*, GR):

$$GR(A_i) = \frac{G(A_i)}{-\sum_{j=1}^{nv(A_i)} \frac{n_{ij}}{n} \log_2 \frac{n_{ij}}{n}}$$

- Problema: cuando n_{ij} tiende a n, el denominador se hace 0

Atributos con costes variables

- Se puede usar

$$\frac{\textit{Ganancia de información}}{\textit{unidad de coste}}$$

o

$$\frac{\textit{Ganancia de información}}{\textit{unidad de coste}^2}$$

- Normalmente, no se llega al árbol de decisión óptimo
- Dominios en los que es útil
  - Medicina (Nuñez, 88)
  - Robótica (Tan y Schlimmer, 90)

Versiones incrementales. ID4 e ID5

- ID4 (Schlimmer y Granger, 86)
  - la reconstrucción del árbol no se hace por cada clasificación errónea, sino por degradación de consistencia
  - mantienen estadísticas de la distribución de los valores de los atributos clasificados debajo de cada nodo
  - si el valor de información de un atributo baja con respecto al de otro, el subárbol se rehace
  - no asume perfecta consistencia con los datos
  - operador de especialización: crecer un árbol
  - operador de generalización: borrar un subárbol
  - eso supone un retroceso simulado
- ID5 (Utgoff, 89)
  - reorganiza los árboles en lugar de borrar y crecer
  - necesita muchos menos ejemplos de entrenamiento

Complejidad

- ID3 crece linealmente con el número de ejemplos de entrenamiento
- ID3 crece exponencialmente con el número de atributos
- El tener más ejemplos no significa que se vaya a mejorar. Hay que seleccionar una ventana. Puede haber ejemplos raros antiguos que sean representativos, pero, normalmente, es mejor escoger los últimos.

Bibliografía

- Machine Learning, Tom Mitchell. Capítulo 3