Árboles de decisión en aprendizaje automático y minería de datos

Tratamiento Inteligente de la Información y Aplicaciones

Juan A. Botía

Departamento de Ingeniería de la Información y las Comunicaciones
Universidad de Murcia

October 4, 2007

Guión de la clase

Introducción

- Métodos de clasificación basados en árboles son frecuentes en
  - estadística, reconocimiento de patrones
  - sobre todo en botánica y diagnóstico médico
- Al ser fáciles de comprender, se tiende a confiar más en ellos que en otros métodos (e.g. redes neuronales)

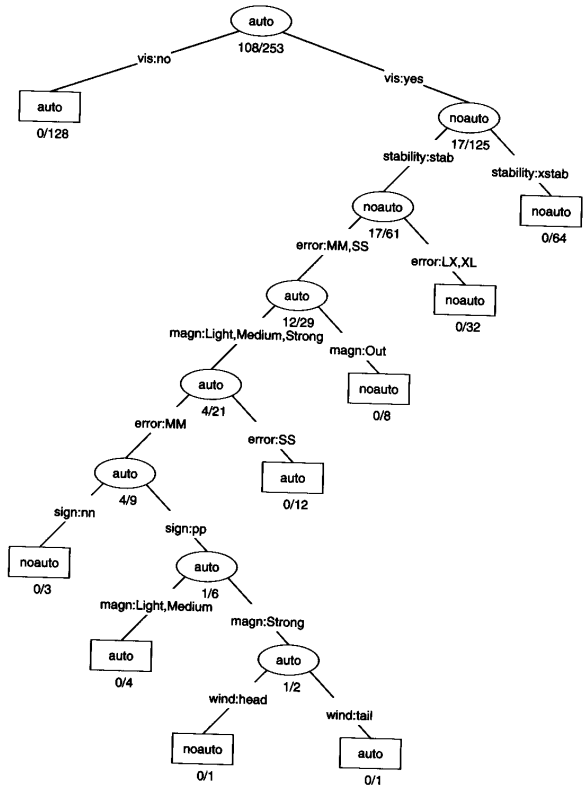
1.	Leaves subterete to slightly flattened, plant with bulb	*2.*
	Leaves flat, plant with rhizome	*4.*
2.	Perianth-tube > 10mm	**I. × hollandica**
	Perianth-tube < 10mm	*3.*
3.	Leaves evergreen	**I. xiphium**
	Leaves dying in winter	**I. latifolia**
4.	Outer tepals bearded	**I. germanica**
	Outer tepals not bearded	*5.*
5.	Tepals predominately yellow	*6.*
	Tepals blue, purple, mauve or violet	*8.*
6.	Leaves evergreen	**I. foetidissima**
	Leaves dying in winter	*7.*
7.	Inner tepals white	**I. orientalis**
	Tepals yellow all over	**I. pseudocorus**
8.	Leaves evergreen	**I. foetidissima**
	Leaves dying in winter	*9.*
9.	Stems hollow, perianth-tube 4–7mm	**I. sibirica**
	Stems solid, perianth-tube 7–20mm	*10.*
10.	Upper part of ovary sterile	*11.*
	Ovary without sterile apical part	*12.*
11.	Capsule beak 5–8mm, 1 rib	**I. enstata**
	Capsule beak 8–16mm, 2 ridges	**I. spuria**
12.	Outer tepals glabrous, many seeds	**I. versicolor**
	Outer tepals pubescent, 0–few seeds	**I. × robusta**

Clasificación en botánica de la especie Iris

Partición jerárquica de los datos

Un árbol de clasificación particiona el espacio sobre el que están definidos los ejemplares de aprendizaje en sub-regiones

stability	error	sign	wind	magnitude	visibility	decision
any	any	any	any	any	no	auto
xstab	any	any	any	any	yes	noauto
stab	LX	any	any	any	yes	noauto
stab	XL	any	any	any	yes	noauto
stab	MM	nn	tail	any	yes	noauto
any	any	any	any	Out of range	yes	noauto
stab	SS	any	any	Light	yes	auto
stab	SS	any	any	Medium	yes	auto
stab	SS	any	any	Strong	yes	auto
stab	MM	pp	head	Light	yes	auto
stab	MM	pp	head	Medium	yes	auto
stab	MM	pp	tail	Light	yes	auto
stab	MM	pp	tail	Medium	yes	auto
stab	MM	pp	head	Strong	yes	noauto
stab	MM	pp	tail	Strong	yes	auto



Construcción de árboles: planteamiento básico

- Se hace al hacer crecer el árbol desde la raiz: se disgrega de manera sucesiva cada una de las hojas hasta alcanzar un nivel de profundidad determinado
- Cuando existe una partición exacta de los datos, la construcción del arbol es muy simple
- Si no existe, tenemos solapamiento de partes en la partición
  - Problema del sobreaprendizaje
  - Soluciones: (1) parar de hacer crecer el árbol antes de desarrollarlo por completo o (2) podar el árbol una vez se ha construido completamente

Construyendo un algoritmo

Podemos decir que existen muchos algoritmos para construcción de árboles de decisión pero que lo que los diferencia virtualmente a todos es

- la regla usada para la disgregación del árbol
- la estrategia de poda

Otras diferencias son

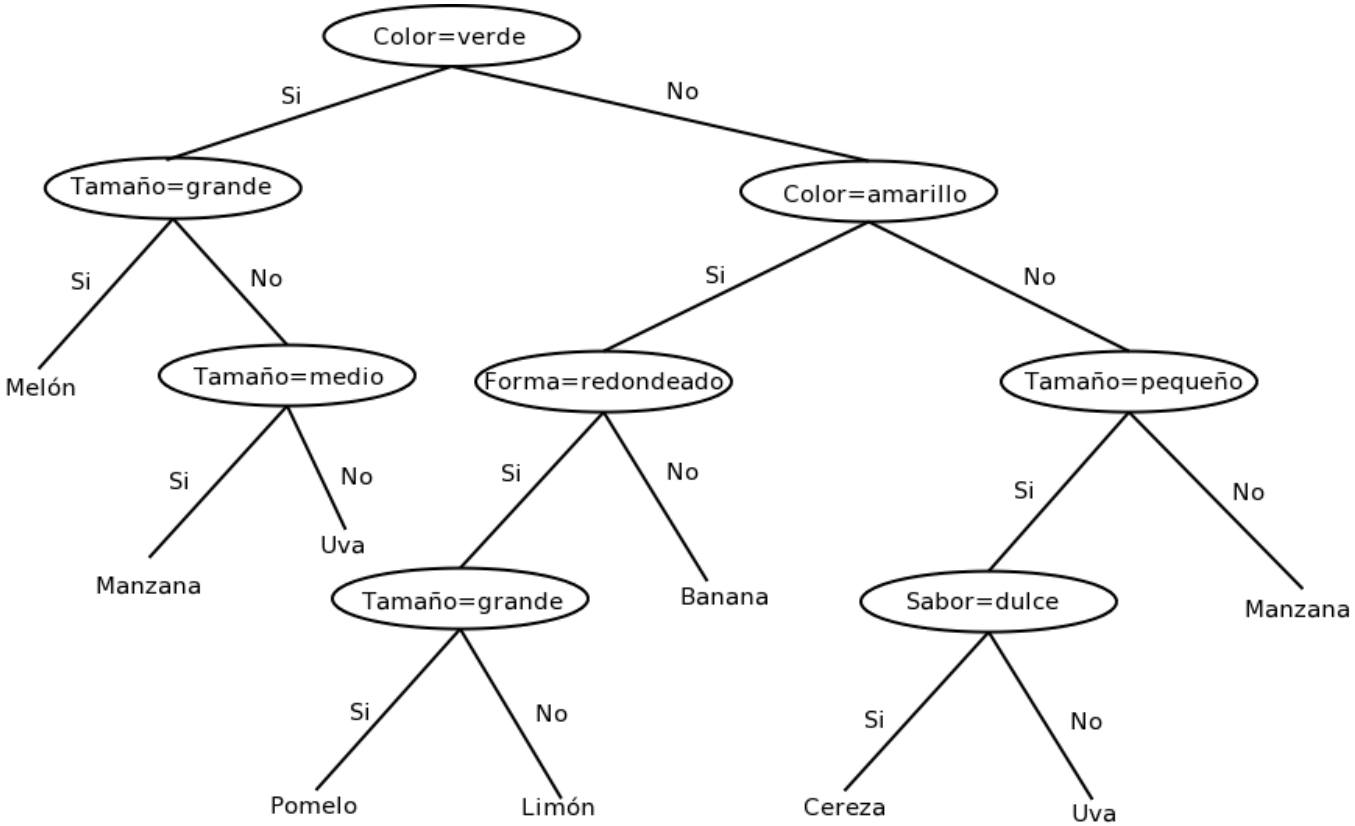
- disgregaciones binarias o n-arias en cada test

Qué optimizar en el problema de búsqueda

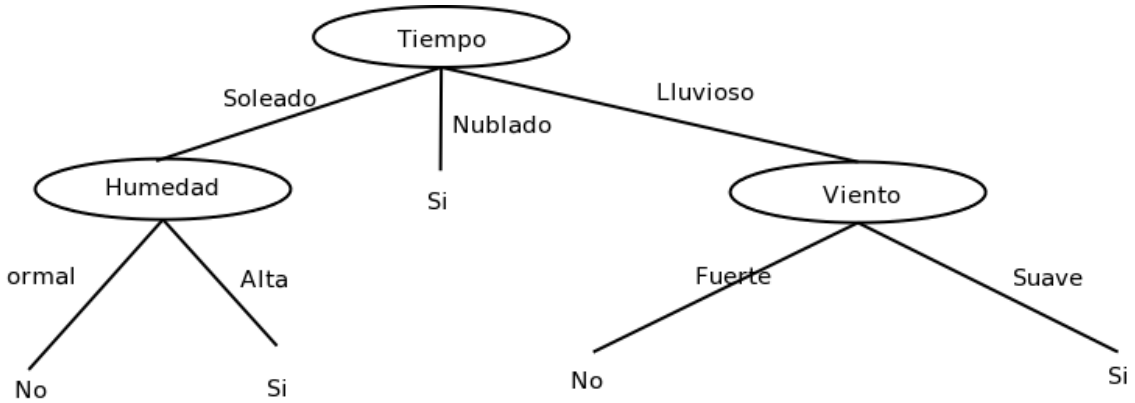
Optimalidad en dos vertientes diferentes

- Buscar la partición óptima de $\mathcal{D}$, en términos de precisión en la clasificación de sus ejemplares
  - Podríamos realizar una búsqueda exhaustiva (impracticable)
- Cómo representar una partición de datos, haciendo uso de árboles, de la manera más conveniente una vez está fijada

Ejemplos → Árbol binario

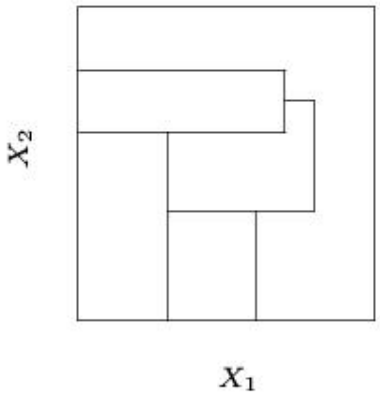


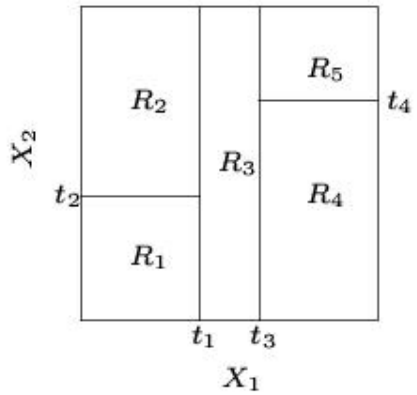
Ejemplos → Árbol n-ario

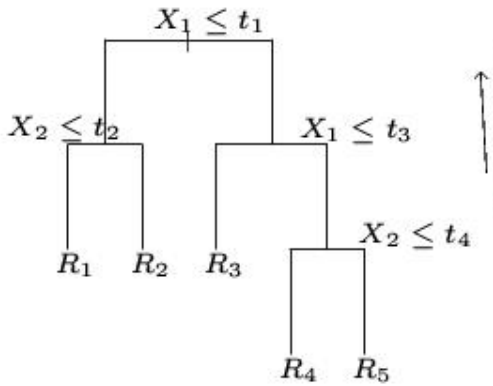


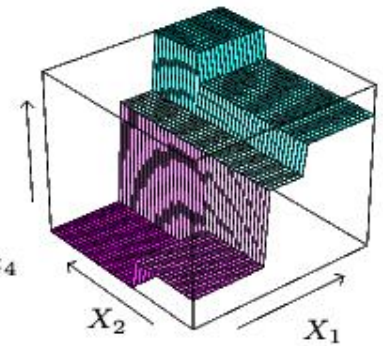
$$
\begin{array}{cc}
 & (Tiempo = Soleado \wedge Humedad = Normal) \\
\vee & (Tiempo = Nublado) \\
\vee & (Tiempo = Lluvioso \wedge Viento = Suave) \\
 & THEN \text{ JUGAR=Si}
\end{array}
$$

Árboles monotéticos - Espacio de hipótesis









Aplicabilidad

- Los ejemplos se presentan en forma de pares

$$< atributo, valor >$$

- Mejor si los atributos tienen un dominio de valores reducido.
- La función de salida presenta valores discretos.
- Es interesante el tipo de representación con vistas a la explotación posterior del modelo.
- Resulta conveniente una representación del tipo de la disyunción de conjunciones.
- Los datos de aprendizaje pueden contener errores.
- Los datos de aprendizaje pueden contener valores nulos en algún atributo para algún ejemplo.
- Ejemplos pueden verse en [2]

El algoritmo básico de aprendizaje

- Estrategia de búsqueda *top-down*, del tipo *greedy*.
- El espacio de búsqueda es el formado por todos los árboles de decisión posibles.
- El algoritmo por excelencia es el `ID3`
- Se comienza respondiendo a

  ¿qué atributo usamos como raíz para el árbol?

- Esta cuestión se resuelve aplicando un test estadístico para averiguar cual de los atributos clasificaría mejor las instancias por sí solo.
- Se desarrolla una rama para cada posible valor.
- En los nuevos nodos se vuelve a hacer la misma pregunta
- Así hasta desarrollar un árbol completo (en la versión básica)

El algoritmo básico de aprendizaje (II)

ID3(Ejemplos, Etiquetas, Atributos)

- Paso 0: Definición
  - Sea `Ejemplos` el conjunto de ejemplos,
  - `Etiquetas` es el conjunto de posibles clases.
  - `Atributos` es el cjto. de atributos en los datos.
- Paso 1: Crear un nodo raíz para el árbol.
- Paso 2: Si todos los ejemplos son positivos, devolver el nodo raíz, con etiqueta $+$.
- Paso 3: si todos los ejemplos son negativos, devolver el nodo raíz, con etiqueta $-$.
- Paso 4: Si Atributos está vacío, devolver el nodo raíz, con el valor de Etiquetas más probable en Ejemplos.
- Si no
  - Inicializar $A \leftarrow$ atributo que *mejor* clasifica Ejemplos.
  - Hacer que el nodo root tenga como atributo de decisión al atributo A.
  - Ahora $\forall v_i \in A$
    - Añadir arco bajo raíz, con test $A = v_i$
    - Sea $Ejemplos_{v_i}$ el subconjunto de Ejemplos con valor v_i en el atributo A.
    - Si $Ejemplos_{v_i} = \emptyset$ añadir un nodo hoja al arco que acabamos de añadir con la etiqueta de Etiquetas más probable en Ejemplos.
    - Sino añadir al nuevo arco el subárbol generado por $ID3(Ejemplos_{v_i}, Etiquetas, Atributos - \{A\})$
- Paso 5: Devolver el nodo raíz

Disgregación de nodos hoja

Dependiendo del número de valores por atributo

- Si los atributos son binarios (i.e. solo toman dos valores diferentes) el tipo de disgregación será binaria (i.e. se generarán dos ramas nuevas a partir de la hoja a disgregar)
- Si los atributos son categóricos con más de dos valores, digamos L, con $L > 2$
  - Podemos considerar disgregaciones binarias hasta haber realizado tests para los L valores
  - También podemos generar sucesivas disgregaciones binarias hasta haber realizado tests para los L valores
- Si los atributos son ordinales, los tests en las disgregaciones serán del tipo $x \leq x_c$
- Combinaciones lineales de atributos continuos y expresiones lógicas

Distribuciones de probabilidad al construir el árbol

En cada hoja, se va a disponer de un conjunto de atributos candidatos para usar en la siguiente disgregación.

- Sea $\mathcal{D} \times C$ el espacio cartesiano formado por los datos de entrada ($\mathcal{D}$) y sus clases ($\mathcal{C}$)
- Sea una hoja del árbol a disgregar
  - Si elegimos el atributo A para disgregar la hoja, supongamos que A tiene los valores posibles $a_1, \ldots, a_m$
  - La distribución de probabilidad sobre esos valores y las clases y la hoja hija correspondiente al test $A = a_i$ viene dada por

$$p(k|a_i) = p_{ik}/p_i$$

a través de las diferentes k clases, siendo p_i sobre $\mathcal{D}$ y p_{ik} sobre $\mathcal{D} \times C$

Medidas de pureza de atributos en la disgreación

Idea básica: la elección del siguiente atributo a disgregar podría basarse en comprobar si el conjunto de los nodos hijos generados a partir de la combinación hoja+atributo son más puros que el padre

Impureza: una medida de impureza ha de ser tal que se haga 0 si la distribución de probabilidad sobre las clases, p_j se concentra en una sola clase y además maximal si p_j es uniforme.

Más sobre impureza en los nodos

Ejemplo

- Si escogemos el atributo `Outlook`, tendremos dos nuevos hijos. En uno de ellos, a_{Sunny}, caerán los ejemplares `D1, D2, D6` y en el otro, a_{Rain}, `D3, D4, D5`. La distribución de probabilidad de las clases ahora es $p_{Sunny,No} = 1$ y $p_{Rain,Yes} = 1$. Por lo tanto, los dos nuevos nodos son más puros que el padre, si este es la raiz.
- Si escogemos el atributo `Temperature`, tendremos tres nuevos hijos. En uno de ellos, a_{Hot}, caerán los ejemplares `D1, D3`, en el segundo, a_{Cool} caerán `D2, D4`. En el tercero, a_{Mild} tendremos los ejemplares `D5, D6`. La distribución de probabilidad de las clases ahora es $p_{Hot,No} = 0.5$ y $p_{Hot,Yes} = 0.5$. Lo mismo para los otros dos valores. Por lo tanto, los tres nuevos nodos son igual de puros que el padre, si este es la raiz.

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Cool	High	Strong	No
D3	Rain	Hot	High	Weak	Yes
D4	Rain	Cool	High	Weak	Yes
D5	Rain	Mild	Normal	Weak	Yes
D6	Sunny	Mild	Normal	Strong	No

Entropía e Indice Gini

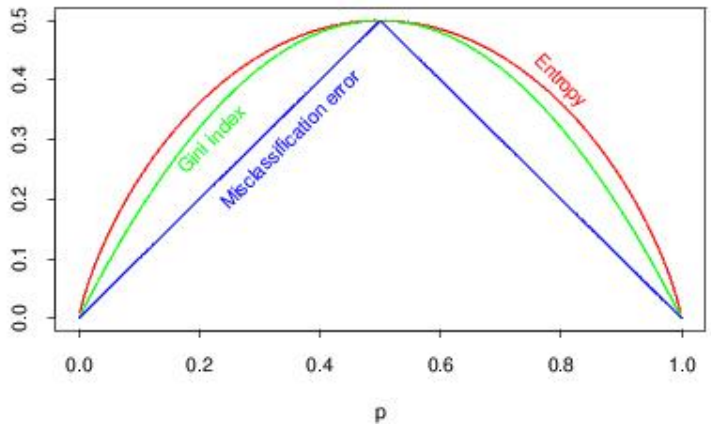
La entropía viene dada por la expresión

$$i(p) = -\sum_j p_j \log pj,$$

El índice Gini es

$$i(p) = \sum_{i \neq j} p_i p_j = 1 - \sum_j p_j^2.$$

El error de clasificación también puede usarse



Medida de disminución de impureza

Entropía y Gini no son suficientes:

> *tenemos que agregar los valores de la función escogida para cada uno de los nodos hijos*

Dado un atributo A, la disminución en impureza promedio (a maximizar) al usar ese atributo en la disgregación vendrá dada por

$$i(p_c) - \sum_{i=1}^{m} p_i \times i(p(c|a_i)),$$

siendo

- $i(p_c)$ la impureza del nodo padre
- $i(p(c|a_i))$ la impureza en el correspondiente nodo hijo generado a partir de un test para el valor a_i de A
- $\sum_{i=1}^{m} p_i \times i(p(c|a_i))$ la impureza generada en los hijos

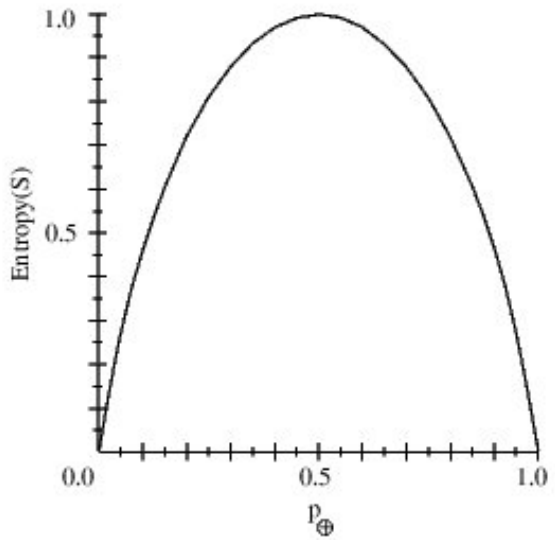
Disgregación en ID3

- Medida básica → **ganancia de información**
- Entropía, Et: la cantidad de bits, en promedio, que harían falta para codificar mensajes que indicaran las clases de los ejemplos. Con ejemplos positivos y negativos...

$$Et(S) = -p_{\oplus} log_2 p_{\oplus} - p_{\ominus} log_2 p_{\ominus}$$

- La función de entropía corresponde a los valores $p_{\oplus} + p_{\ominus} = 1$
- Si una clase tiene $P = 1$ entropía es 0.
- Valor máximo → $p_{\oplus} = p_{\ominus} = 0.5$.
- Si la etiqueta de los ejemplos puede tomar c valores diferentes

$$Et(S) = \sum_{i=1}^{c} -p_i log_2 p_i$$



Ganancia de información

- Informalmente es la reducción en entropía del conjunto, al clasificar S usando el ejemplo determinado.
- Es una medida relativa al conjunto S y a cada atributo.

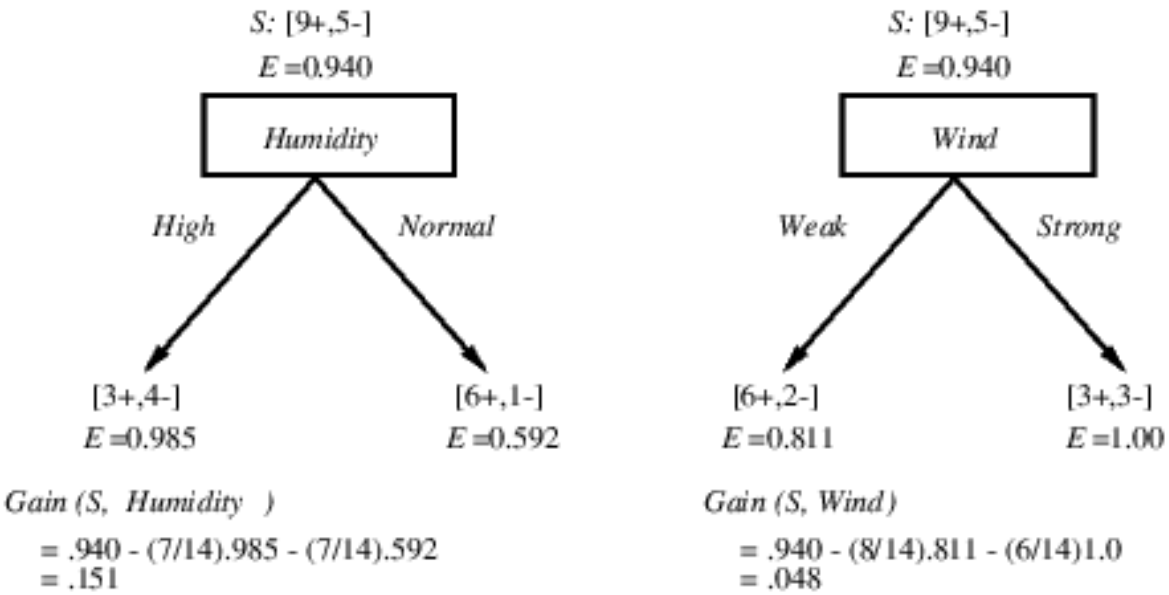
$$Ganancia(S, A) = Et(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} Et(S_v)$$

  - $Valores(A)$ es el conjunto de posibles valores del atributo A,
  - $|S_v|$ es el número de ejemplos en S etiquetados con v,
  - $|S|$ es el número total de ejemplos y
  - $Et(S_v)$ es la entropía de los ejemplos etiquetados con v.

Ejemplo

Data	Outlook	Temperature	Humidity	Wind	PlayTennis
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

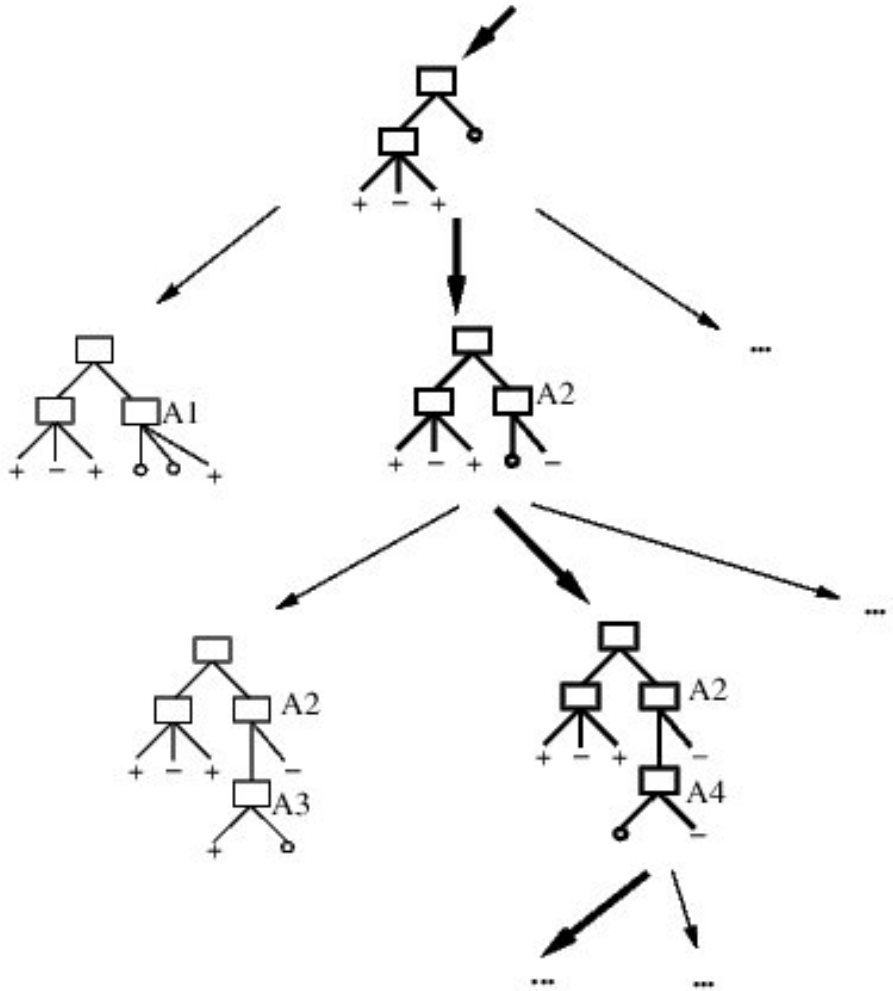
Ejemplo


S: [9+,5-]
E =0.940
Humidity
High
Normal
[3+,4-]
E =0.985
[6+,1-]
E =0.592
Gain (S, Humidity)
= .940 - (7/14).985 - (7/14).592
= .151
S: [9+,5-]
E =0.940
Wind
Weak
Strong
[6+,2-]
E =0.811
[3+,3-]
E =1.00
Gain (S, Wind)
= .940 - (8/14).811 - (6/14)1.0
= .048

Comparativa de medidas

- La medida de error de clasificación no es diferenciable
- Entropía y Gini son más precisas al reflejar cambios en probabilidades de nodos
  - Sea un problema binario con 400 ejemplos en cada clase (400,400)
  - Supongamos un par de posibles splits
    1. (300,100) y (100,300)
    2. (200,400) y (200,0)
  - Los dos tienen un error de clasificación de 0.25
  - Entropía y Gini son inferiores en el segundo

Búsqueda en ID3



Búsqueda en árboles de decisión

- El espacio de búsqueda es completo (el árbol que buscamos está ahí)
- El tipo de búsqueda es *top-down* con estrategia `hill-climbing` (o greedy) por lo que no hay backtracking (mínimos locales)
- Búsqueda basada en probabilidades (robustez al ruido)
- Tendencia en la búsqueda: preferencia por los árboles con caminos a las hojas más cortos desde la raiz

Sobre-aprendizaje

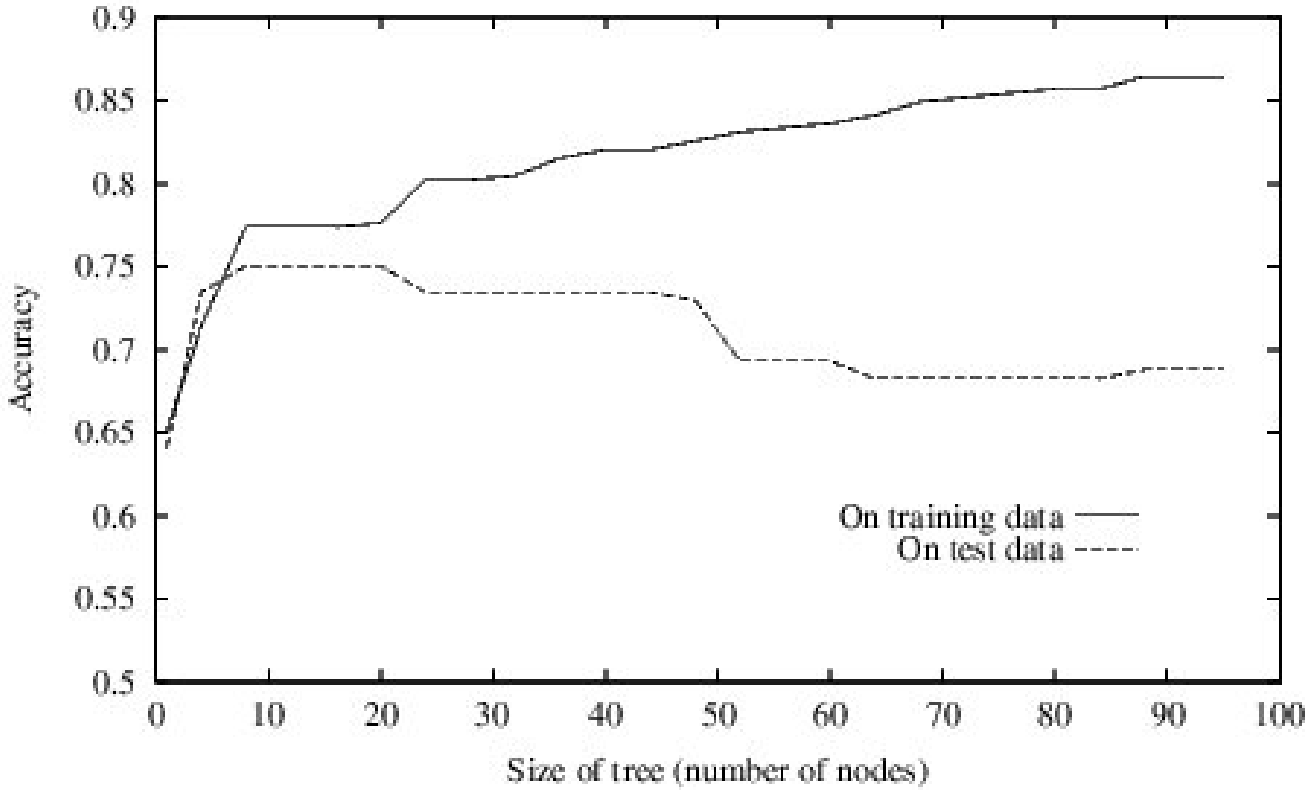
`ID3` puede adolecer de *overfitting*.

- El conjunto de ejemplos no es lo suficientemente representativo
- Los ejemplos tienen errores

Definición

Dado un espacio de hipótesis H, se dice que una hipótesis particular $h \in H$ sobreajusta los datos de entrenamiento si existe una hipótesis alternativa $h' \in H$, tal que h presenta un error menor que h' sobre los ejemplos de entrenamiento, pero h' presenta un error menor que h sobre el conjunto total de observaciones.

Sobre-aprendizaje (II)



Influencia de errores en el *overfitting*

- Conjunto de ejemplos de la aplicación inicial

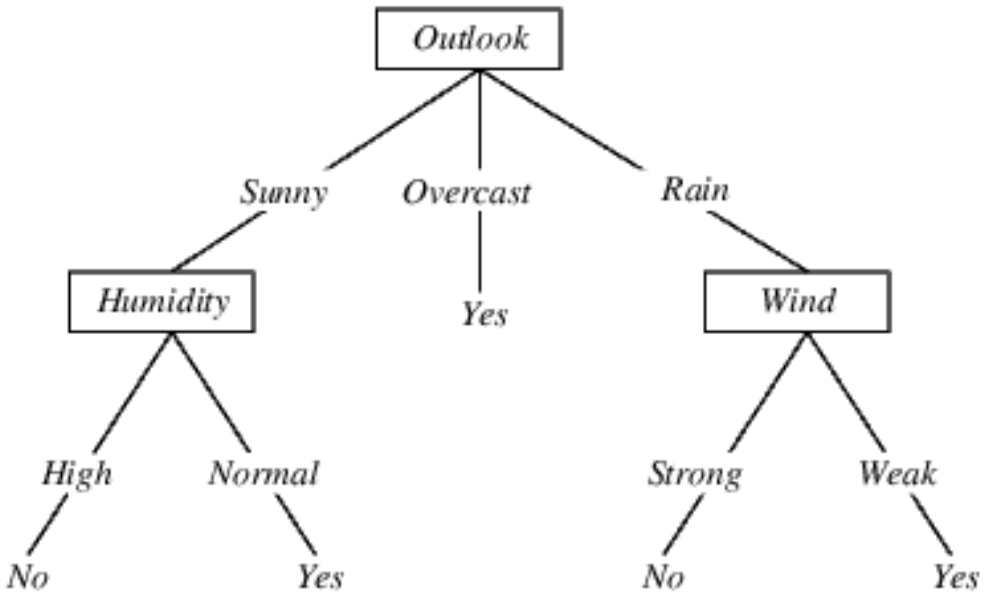
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	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

- Introducimos un nuevo ejemplo

$$< Outlook = Sunny,\ Temperature = Hot,\ Humidity = Normal,\\ Wind = Strong,\ PlayTennis = No >$$

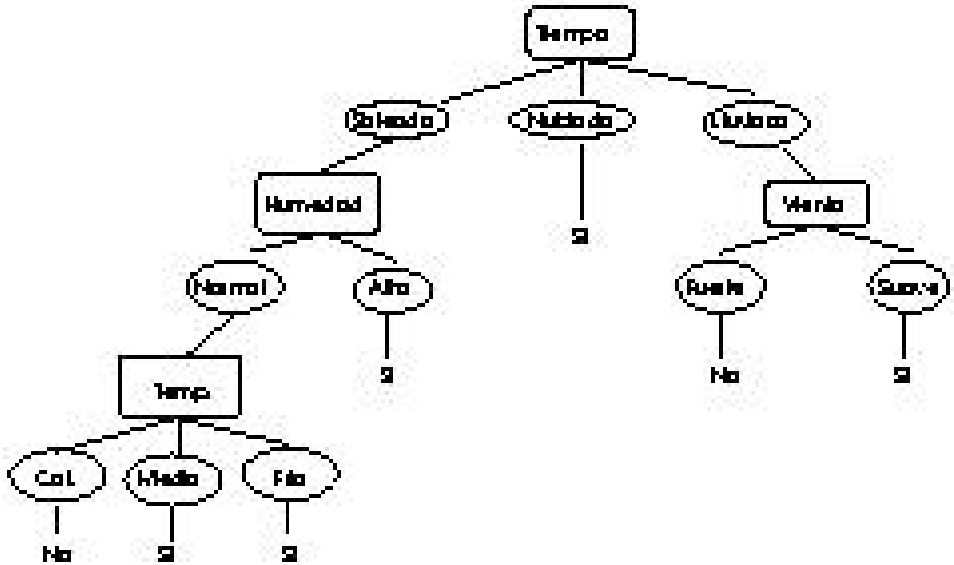
Influencia de errores en el *overfitting* (y II)

El árbol antes era



Influencia de errores en el *overfitting* (y III)

Se generará un nuevo nodo en el árbol para asimilar ese ejemplo



El nuevo árbol h es más complejo y preciso en el error de entrenamiento que h', aunque es de esperar que generalice peor

Encontrando el tamaño más adecuado para el árbol

¿Cómo podemos evitar el sobreaprendizaje?

- Parar de construir el árbol, antes de que este clasifique perfectamente todos los ejemplos (*pre-prunning*).
- Se puede construir el árbol y después intentar una poda (*post-prunning*).

Incorporación de valores continuous

- Transformamos el dominio de los continuos en una serie de intervalos
  - Sea A un atributo continuo,
  - El nuevo A_c será *true* si $A < c$ y *false* si $A \geq c$.
  - Problema: umbral c.
- Nuevo atributo, `Temperatura`

Temperatura	40	48	60	72	80	90
Jugar Tenis	No	No	Si	Si	Si	No

- Podemos usar la medida de impureza para decidir entre varios posibles valores para c
  1. Candidato: $Temp_{54} = (60 + 48)/2$ y
  2. Candidado: $Temp_{85} = (80 + 90)/2$.
  3. $Ganancia(Temp_{54}) > Ganancia(Temp_{85})$

Otras medidas alternativas para selección de atributos

- $Ganancia(S, A)$ favorece a los atributos que tienen más valores.
- Ejemplo: atributo `fecha`.
- Solución: nueva medida que tenga en cuenta este hecho

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

  Es, en realidad, la entropía del conjunto S con respecto a los valores del atributo A.
- El ratio de ganancia vendrá dado por la expresión

$$GainRatio(S, A) = \frac{Ganancia(S, A)}{SplitInformation(S, A)}$$

- Por tanto, se amortigua la elección de atributos con muchos valores, uniformemente distribuidos.

Ejemplos de entrenamiento con valores nulos

- Es posible que en S tengamos valores desconocidos
- Al calcular $Ganancia(S, A)$, siendo $< x, c(x) >$ un ejemplo de S para el cual el valor $A(x)$ no se conoce. ¿Cómo damos un valor a $A(x)$? Opciones
  - Calcular el valor más probable
  - Calcular el valor más probable de entre los ejemplos que pertenecen a la clase $c(x)$.
  - Asignar probabilidades a los distintos valores de un determinado atributo.
    - Sea A un atributo booleano.
    - Se observan en S 6 valores de verdad `true` y
    - 4 valores de verdad `false`.
    - Para nuevos $< x, c(x) >$ con valor nulo para A le asignaremos un `true` con probabilidad 0.6 y `false` con probabilidad 0.4.

Atributos con relevancias diferentes

- Podemos necesitar diferenciar atributos en términos de su coste
- Ejemplo, diagnóstico médico con atributos `Temperatura`, `ResultadoBiopsia`, `Pulso` y `ResultadoTestSanguíneo`.
- Ejemplos de medidas

$$\frac{Ganancia^2(S,A)}{Coste(A)}$$

$$\frac{2^{Ganancia(S,A)}-1}{(Coste(A)+1)^w},\ w \in [0,1]$$

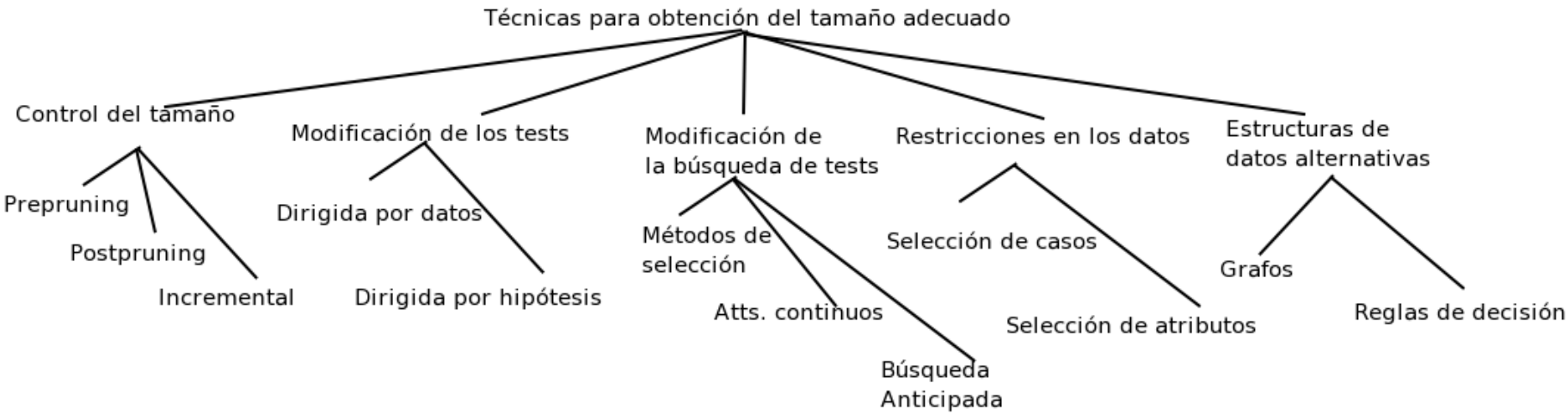
Obtención de un AD con el tamaño adecuado

Motivación

Construimos un conjunto de datos artificialmente con, digamos, 10 atributos, cada uno de los cuales tomando valores en $\{0, 1\}$*, con igual probabilidad, dos clases diferentes,* `yes` *con probabilidad 0.25 y* `no` *con probabilidad 0.75. Generamos 1000 ejemplos, y los dividimos aleatoriamente en 500 ejemplos de test y 500 ejemplos de entrenamiento. Pues bien, el algoritmo básico produce un árbol de 119 nodos, con un error de clasificación del 35% en los casos de test. Obsérvese que un árbol con una única hoja etiquetada con la clase* `no` *habría producido un error de clasificación del 25%.*

1. Un árbol balanceado, y con un número pequeño de nodos es más fácil de analizar
2. "pequeños disjuntos" $\longrightarrow$ clasifican pocos casos
3. Ruído nos lleva a la sobreadaptación

Técnicas de control de crecimiento [1]



Técnicas (II)

Control del tamaño: en este enfoque se intenta controlar el tamaño, bien intentando construir un árbol limitado, podándolo ulteriormente ó ajustandolo on-line.

- *Pre-prunning*: imposición de un criterio, no trivial, con el que parar de expandir el árbol.
- *Post-prunning*: eliminación de subárboles después de la construcción del árbol total.
- *Reajuste incremental*: si se mantienen los casos de entrenamiento, y se van acumulando los subsiguientes casos que vayan apareciendo el árbol puede ir actualizándose on-line.

Técnicas (III)

Modificación del espacio de los tests

- La construcción de test dirigida por datos se basa en construir nuevos atributos mediante
  1. combinación de atributos base mediante operadores numéricos,
  2. por combinación de éstos por operadores lógicos
- En la construcción de tests dirigida por hipótesis
  - se almacenan los tests construidos según la forma anterior,
  - los árboles que van construyéndose influyen en la decisión de si aplicarlos postreramente o no.

Técnicas (IV)

- Modificación de la búsqueda de tests: se puede modificar la búsqueda usando una diferente a la de tipo *greedy*.
  - Nuevas medidas de selección: podemos usar medidas alternativas a la ganancia como por ejemplo la basada en el principio MDL (*Minimun Description Length*), la medida de Kolmogorov-Smirnoff, separabilidad de clases, etc.
  - Uso de atributos continuos: la discretización de atributos continuous se puede realizar de tal forma que se reduzca el sesgo y la selección de tests en atributos continuous.
  - Búsqueda anticipada: en esta, se construyen en forma tentativa subárboles usando determinados tests, y se evalúan a una profundidad suficiente. Así, se puede escoger el tests que mejores resultados ha dado con más información que si se realizara una simple búsqueda *greedy*.

Técnicas (V)

Restricciones en los datos

- Podemos eliminar casos o atributos

Estructuras de datos alternativas

- Una vez se ha construido el árbol, podemos convertir este en una estructura diferente, más compacta
- grafos y
- reglas de decisión


L. A. Breslow and D. W. Aha.
Simplifying decision trees: a survey.
Knowledge Engineering Review, 12(1):1–40, 1997.


Sheerama K. Murthy.
Automatic construction of decision trees from data: A multi-disciplinary survey.
Data Mining and Knowledge Discovery, 1997.