

Aprendizaje Automático Árboles de decisión

Viviana Cotic
1er cuatrimestre 2019



Árboles de decisión

- método para **inferencia inductiva**
- aprenden **reglas if-then** sobre los valores de los atributos. Predicen valor objetivo en función de las reglas.

Árboles de decisión - Ejemplo

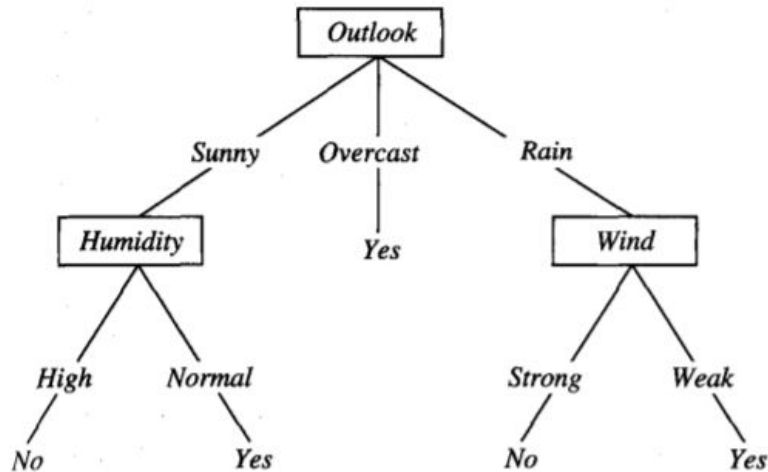


Árboles de decisión - Ejemplo



- . **nodo** representa pregunta
- . **ejes** representan posibles respuestas
- . **hojas:** nodos que representan objetos
- . **caminos desde la raíz.**

Árboles de decisión



. **nodo** representa test sobre un atributo de la instancia

. **rama desde el nodo**: corresponde a un valor para ese atributo.

representa **disyunción de conjunciones sobre valores de atributos**

$(\text{Outlook} = \text{Sunny} \wedge \text{Humidity} = \text{Normal}) \vee$

$(\text{Outlook} = \text{Overcast}) \vee$

$(\text{Outlook} = \text{Rain} \wedge \text{Wind} = \text{Weak})$

Cuándo considerar árboles de decisión

- **Instancias representables por pares atributo-valor** (especialmente pocos valores disjuntos). Veremos con valores continuos
- **La función objetivo tiene valores de salida discretos.** También podrían ser reales, pero es menos común
- **Se pueden requerir hipótesis disyuntivas.**
- **Posible conjunto de entrenamiento ruidoso (noisy)**
- **Posibles valores de atributos faltantes**

Ejemplos de uso:

- diagnósticos médicos
- análisis de riesgo crediticio

Índice

- Árboles de decisión
- **Algoritmo**
 - criterio de selección
 - sesgo inductivo
 - Occam's Razor
 - sobreajuste
 - poda
- Adecuación a valores continuos
- Valores faltantes
- Atributos con costo
- Resumen

Ejemplo

Instancia	Atributos				Clase
	Cielo	Temperatura	Humedad	Viento	Va a correr?
1	sol	calor	alta	débil	No
2	sol	calor	alta	fuerte	No
3	nublado	calor	alta	débil	Sí
4	lluvia	templado	alta	débil	Sí
5	lluvia	frío	normal	débil	Sí
6	lluvia	frío	normal	fuerte	No
7	nublado	frío	normal	fuerte	Sí
8	sol	templado	alta	débil	No
9	sol	frío	normal	débil	Sí
10	lluvia	templado	normal	débil	Sí
11	sol	templado	normal	fuerte	Sí
12	nublado	templado	alta	fuerte	Sí
13	nublado	calor	normal	débil	Sí
14	lluvia	templado	alta	fuerte	No

Cantidad de positivos

Cantidad de negativos

Inducción Top-Down de árboles de decisión

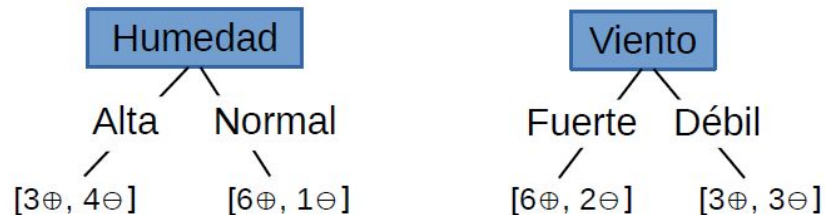
ID-3 y C4.5 (Quinlan)

- 1) $A \leftarrow$ el “mejor” atributo para nodo_actual
- 2) Asignar A como atributo de decisión del nodo_actual
- 3) Para cada valor de A, crear un nuevo hijo del nodo_actual
- 4) Clasificar (repartir) las instancias en los nuevos nodos, según el valor de A
- 5) Si las instancias están bien clasificadas: TERMINAR
Si no: Iterar sobre nuevos nodos

¿Qué atributo es el mejor?

- ☐ Information gain
- ☐ Impureza Gini
- ☐ Gain ratio
- ☐ ...

Tenemos 14 instancias: $[9\oplus, 5\ominus]$



Verificar cuán bien un atributo separa a los ejemplos de acuerdo a su clasificación objetivo.

¿Qué atributo es el mejor?

Medidas de **impureza** de un conjunto de ejemplos:

- Entropía (Information Gain)
- Gini (Gini Gain)

Medidas de efectividad de un **atributo** para clasificar datos de entrenamiento

- **Information Gain:** reducción esperada de entropía por partir ejemplos basados en ese atributo.
- **Gini Gain:** reducción de índice Gini por partir ejemplos basados en ese atributo

¿Qué atributo es el mejor?

Opción 1: Information Gain

Entropía de una muestra S (ejemplos de entrenamiento)

$$\text{Entropy}(S) = \sum_{c \in \text{Clases}} -p_c \log_2 p_c$$

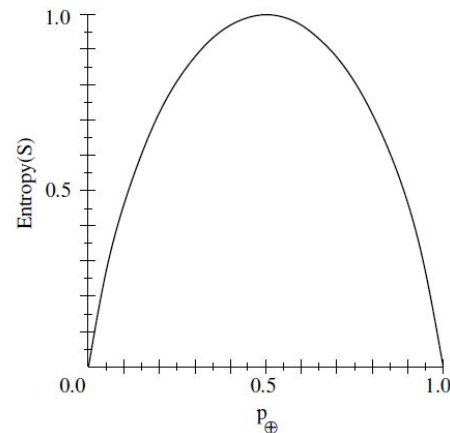
p_c : proporción de instancias de S pertenecientes a clase c

Entropía mide **impureza de S**

Para el caso binario ($c=2$)

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

En ej: $\text{Entropy}([9+, 5-]) = -(9/14) \log_2 (9/14) - (5/14) \log_2 (5/14) = 0.940$



¿Qué atributo es el mejor?

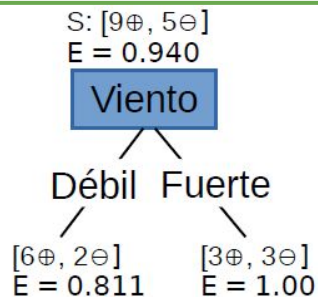
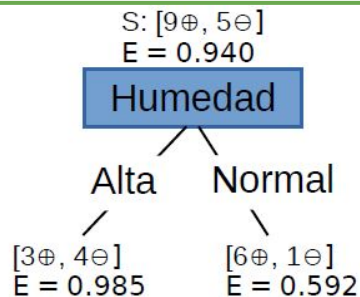
Opción 1: Information Gain

- Reducción de entropía de la muestra S causada por particionar ejemplos de acuerdo a un atributo A

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Valores}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

Valores (A): posibles valores del atributo A

$$S_v = \{s \in S | A(s) = v\}$$



$$\begin{aligned} \text{Gain}(S, \text{Viento}) &= \text{Entropy}(S) - \\ & (8/14) \text{Entropy}(S_{\text{Débil}}) - (6/14) \\ & \text{Entropy}(S_{\text{Fuerte}}) = 0.940 - (8/14) \\ & 0.811 - (6/14) 1 = \mathbf{0.048} \end{aligned}$$

$$\begin{aligned} \text{Gain}(S, \text{Humedad}) &= \text{Entropy}(S) - \\ & (7/14) \text{Entropy}(S_{\text{Alta}}) - (7/14) \\ & \text{Entropy}(S_{\text{Normal}}) = 0.940 - (7/14) \\ & 0.985 - (7/14) 0.592 = \mathbf{0.151} \end{aligned}$$

- Gain Ratio (otra métrica).** Corrige preferencia de Information Gain sobre atributos con muchos valores.

¿Qué atributo es el mejor?

Opción 2: Gini

Se mide impureza de la muestra (conjunto de entrenamiento)

- Impureza Gini

- Impureza de S:

$$Gini(S) = 1 - \sum_{c \in Clases} \left(\frac{|S_c|}{|S|} \right)^2$$

S_c : conjunto de instancias que pertenecen a clase c

- Reducción de impureza de S , respecto a atributo A

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

$Valores(A)$ es el conjunto de valores posibles del atributo A .

$$S_v = \{s \in S \mid A(s) = v\}$$

- Se elige atributo con mayor GiniGain

Construcción usando Information Gain

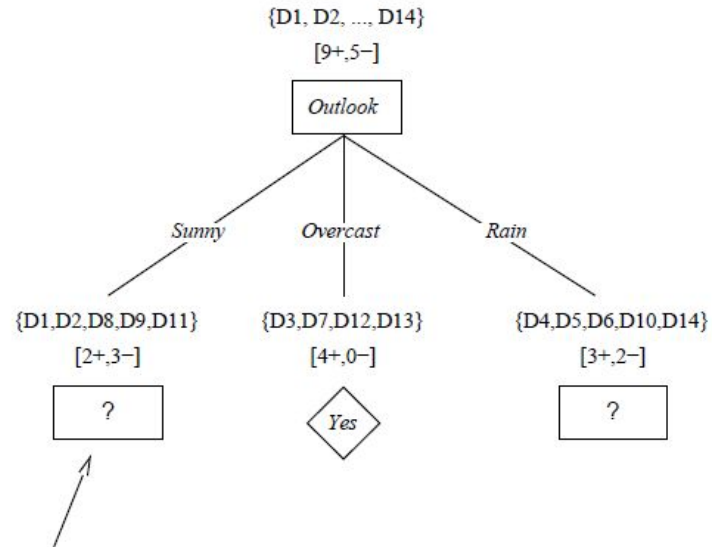
En nuestro ejemplo

Gain (S, Cielo) = 0.246

Gain (S, Humedad) = 0.151

Gain (S, Viento) = 0.048

Gain (S, Temperatura) = 0.029



Which attribute should be tested here?

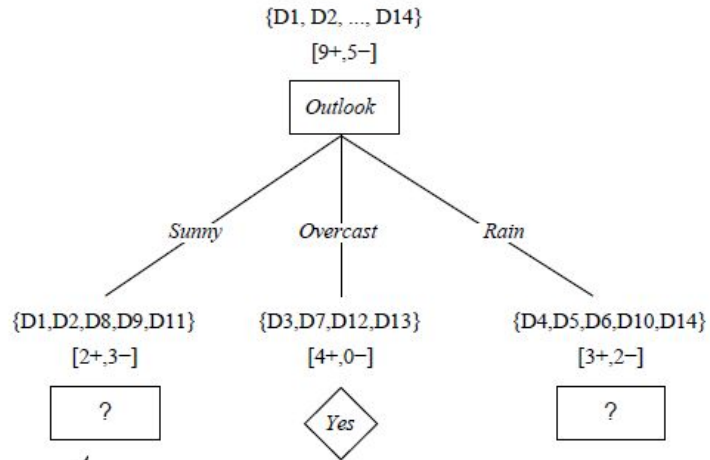
$$S_{\text{Sunny}} = \{D1,D2,D8,D9,D11\}$$

$$\text{Gain}(S_{\text{Sunny}}, \text{Humidity}) = .970 - (3/5) 0.0 - (2/5) 0.0 = .970$$

$$\text{Gain}(S_{\text{Sunny}}, \text{Temperature}) = .970 - (2/5) 0.0 - (2/5) 1.0 - (1/5) 0.0 = .570$$

$$\text{Gain}(S_{\text{Sunny}}, \text{Wind}) = .970 - (2/5) 1.0 - (3/5) .918 = .019$$

Construcción usando Information Gain



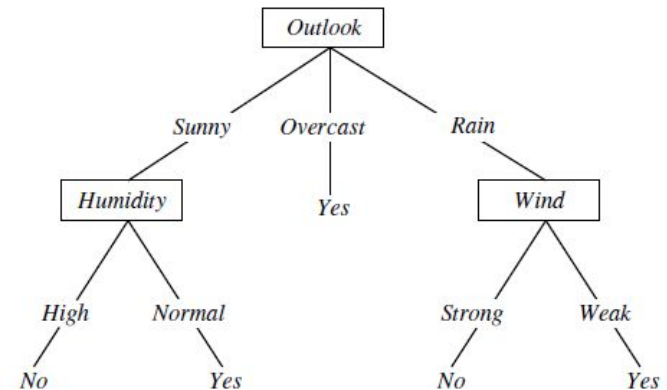
Which attribute should be tested here?

$$S_{\text{Sunny}} = \{D1, D2, D8, D9, D11\}$$

$$\text{Gain}(S_{\text{Sunny}}, \text{Humidity}) = .970 - (3/5) 0.0 - (2/5) 0.0 = .970$$

$$\text{Gain}(S_{\text{Sunny}}, \text{Temperature}) = .970 - (2/5) 0.0 - (2/5) 1.0 - (1/5) 0.0 = .570$$

$$\text{Gain}(S_{\text{Sunny}}, \text{Wind}) = .970 - (2/5) 1.0 - (3/5) .918 = .019$$



Búsqueda en espacio de hipótesis

- **Espacio de hipótesis:** completo (a diferencia de Algoritmo de Eliminación de candidatos -CEA-)
- **Salida:** una sola hipótesis (a diferencia de CEA)
- **No se hace backtracking:** puede haber óptimos locales no globales
- La **búsqueda** se hace **en base a estadísticas:** robusto a errores en datos de entrenamiento
- **Bias inductivo:** se prefiere el árbol más corto

Sesgo inductivo (bias inductivo)

Hay muchos posibles árboles para un conjunto de datos de entrenamiento. ¿Cómo se elige una de las hipótesis consistentes por sobre las otras?

- Preferencia por:
 - árboles más bajos y
 - con atributos con Information Gain alto cerca de la raíz
- Sesgo:
 - **preferencia: búsqueda incompleta** en **espacio de hipótesis completo**. Sesgo: consecuencia de orden de hipótesis de acuerdo a **estrategia de búsqueda. (preferencia de una hipótesis sobre otras)**. Ej: ID3
 - **restricción: búsqueda completa** en **espacio de hipótesis incompleto**. Encuentra todas las hipótesis consistentes con datos de entrenamiento. Sesgo: consecuencia de poder expresivo de la representación de hipótesis. Ej: CEA
- **Navaja de Occam**: se prefiere la hipótesis más corta que satisface a los datos

Navaja de Occam/Ockham (Occam's razor)

Ockham. Filósofo y teólogo (1287-1347)

- *“Pluralitas non est ponenda sine necessitate.”* La pluralidad no debe postularse sin necesidad
- **Occam's razor:** Preferir la hipótesis más simple que se ajuste a los datos.
(Las soluciones simples tienen mayor probabilidad de ser correctas que las complejas.)
- No es un principio irrefutable.

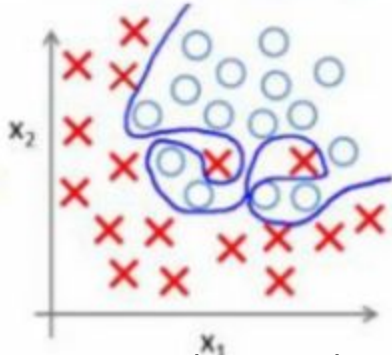
Funes el memorioso - Jorge L. Borges, Ficciones

Fragmentos

“Era casi incapaz de ideas generales, platónicas. No sólo le costaba comprender que el símbolo genérico 'perro' abarcara tantos individuos dispares de diversos tamaños y diversa forma; le molestaba que el perro de las tres y catorce (visto de perfil) tuviera el mismo nombre que el perro de las tres y cuarto (visto de frente).”

“Había aprendido sin esfuerzo el inglés, el francés, el portugués, el latín. Sospecho, sin embargo, que no era muy capaz de pensar. Pensar es olvidar diferencias, es generalizar, abstraer. ”

Overfitting - Sobreajuste



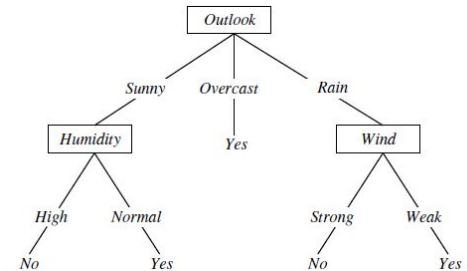
Fernando Berzal, DECSAI,
Universidad de Granada

En árboles de decisión, hay sobreajuste cuando el árbol es “demasiado” profundo

¿Qué pasa si hay **descripciones exactas de instancias únicas y aisladas**?

Ej, si agregamos este caso erróneo a nuestro árbol

15. (Sol, Calor, Normal, Fuerte) No



Overfitting

Considerar el error de una hipótesis o modelo h sobre:

- instancias de entrenamiento (D): $\text{error}_D(h)$
- todas las instancias posibles (X): $\text{error}_X(h)$

Definición: Una **hipótesis h sobreajusta** a los datos de entrenamiento **si existe** otra hipótesis **h'** , tal que:

$$\text{error}_D(h) < \text{error}_D(h') \text{ y}$$

$$\text{error}_X(h) > \text{error}_X(h')$$

Es decir, h es mejor sobre cjto. entrenamiento, pero h' generaliza mejor

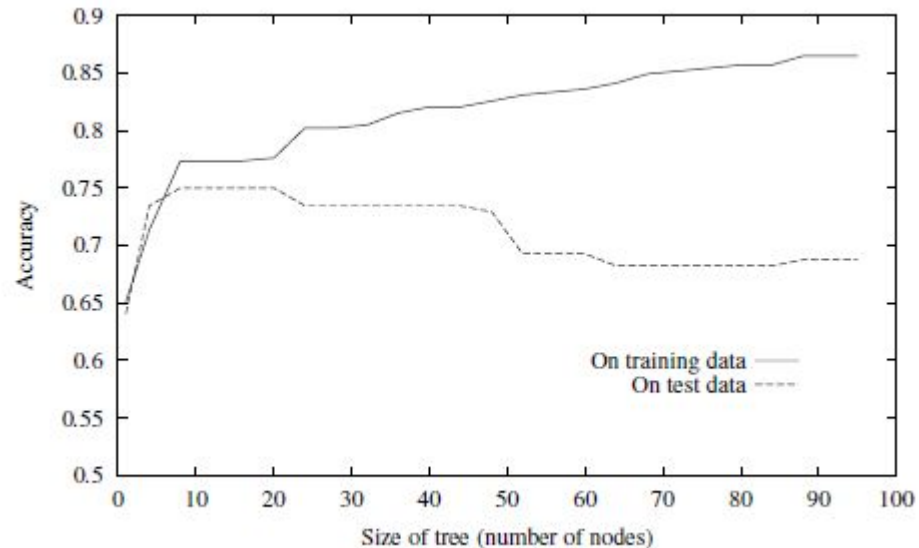
Overfitting

Datos:

- entrenamiento
- test (independiente)

Accuracy (exactitud):

$$(TP+TN)/(TP+TN+FP+FN)$$



Machine Learning, Tom M. Mitchell, McGrawHill, 1997

Overfitting en Árboles de Decisión - Cómo evitarlo

Soluciones:

- **detener crecimiento del árbol** antes de que clasifique perfectamente a los datos
- hacer crecer el **árbol entero**, luego **podar (post-prune)**

Overfitting en Árboles de Decisión - Cómo evitarlo

Cuál es el tamaño correcto del árbol?

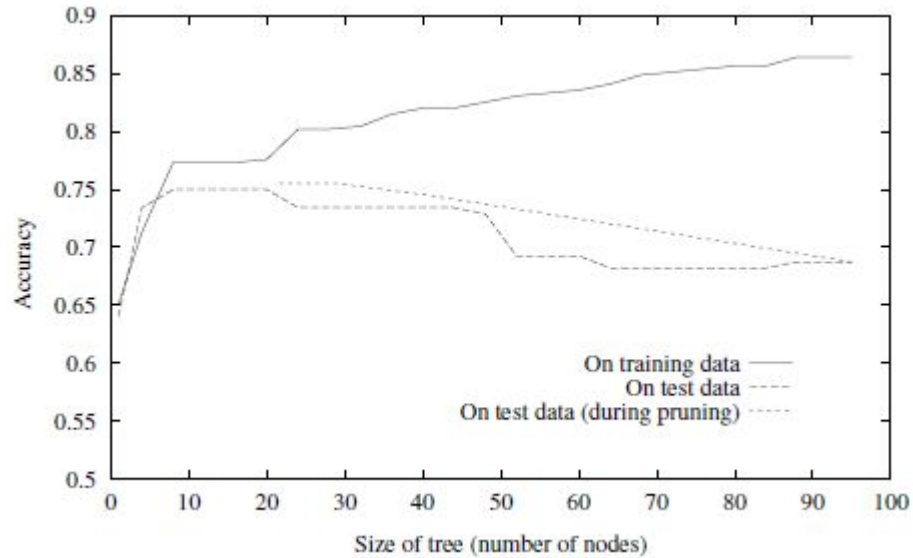
- Conjuntos de entrenamiento y validación:** usar otros datos distintos al entrenamiento para evaluar utilidad de post-pruning
- Conjunto de Entrenamiento y test estadístico:** usar todos los datos para entrenamiento y aplicar test estadístico para evaluar si expandir o podar mejora resultados con datos de entrenamiento
- Minimum Description Length (MDL):** uso de medida de complejidad basada en ejemplos de entrenamientos y árbol de decisión. Se detiene crecimiento al minimizarse.

Reduced Error Pruning

Uso de conjunto de validación. Se considera que cada nodo es candidato sde pruning.

1. Particionar datos en conjuntos de entrenamiento y validación
 2. Repetir hasta que poda sea perjudicial
 - a. Evaluar impacto (en conjunto de validación) de podar cada posible nodo (y todos los de abajo). (Remover todo el subárbol, convertirlo en hoja. Asignación de clasificación más habitual según conjunto de entrenamiento)
 - b. Remover aquél que mejora el accuracy del validation set (greedy o ansioso)
- Produce tamaño menor del subárbol más accurate (exacto)

Reduced Error Pruning



Machine Learning, Tom M. Mitchell, McGrawHill, 1997

Rule Post Pruning

1. Inferir el árbol, de forma tal que satisfaga el conjunto de entrenamiento (con posible sobreajuste)
2. Convertir el árbol en conjunto de reglas (una regla para cada camino desde raíz hasta la hoja)
3. Podar cada regla independientemente de las demás. Removiendo precondiciones que mejoran la accuracy
4. Ordenar las reglas de 3 de acuerdo a su accuracy estimada. Usarlas en ese orden al estimar nuevos datos.

Rule Post Pruning

2. Conversión de árbol a reglas

IF Outlook = "Sunny" \wedge Humidity = "High" **THEN** Corre = No



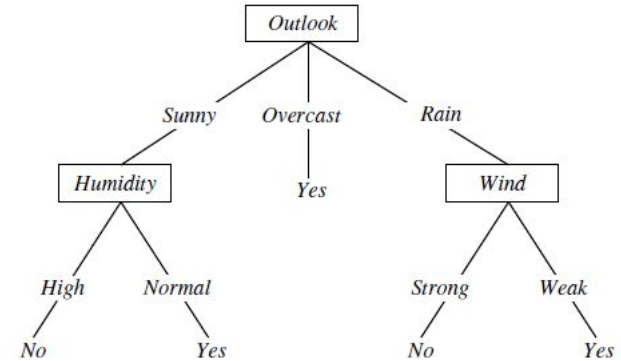
antecedente o precondition



consecuente o postcondición

IF Outlook = "Sunny" \wedge Humidity = "Normal" **THEN** Corre = Yes

...



Usado por C4.5 (Quinlan 1993)

3. si Accuracy de eliminar algún antecedente mejora, se remueve
se prueba sacando el 1ro y el 2do

Se puede usar validation set o training set (con un estimador pesimista para compensar el hecho de usar training set)

Índice

- Árboles de decisión
- Algoritmo
- **Adecuación a valores continuos**
- Valores faltantes
- Atributos con costo
- Resumen

Atributos de valores continuos

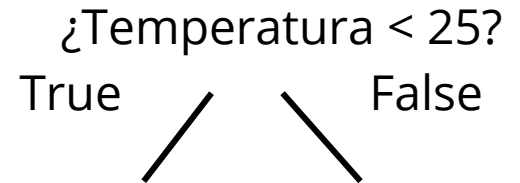
Si tenemos A, un atributo numérico, **discretizamos**. Definimos nuevos atributos discretos que particionan los valores de A en intervalos discretos.

Buscamos un umbral t y discriminamos en función de si $A < t$.

¿Cómo elegir t?:

- 1) se ordenan instancias de menor a mayor A
- 2) se busca forma de partir la lista, de forma tal que maximice la reducción de impureza (umbrales posibles: 6,11,25,33), ej. $(22+28)/2$

Temperatura	3	5	7	15	22	28	32	34
¿Corre?	Sí	Sí	No	Sí	Sí	No	No	Sí



Existen **extensiones** para particionar atributos continuos en múltiples intervalos

Índice

- Árboles de decisión
- Algoritmo
- Adecuación a valores continuos
- **Valores faltantes**
- Atributos con costo
- Resumen

Atributos con valores faltantes

Ej: faltan datos de algunos atributos de algunas de nuestras instancias

- Se estima el valor faltante en base a otros ejemplos para los cuáles el atributo tiene un valor.
- Quiero definir $\text{Gain}(S, A)$ en nodo n para ver si A es el mejor atributo a testear $\langle x, c(x) \rangle$ instancia y el valor $A(x)$ es desconocido

Posibles estrategias:

- asignar el **valor más común** entre los datos de **entrenamiento**
- asignar el **valor más común** entre los datos de **entrenamiento** que tienen la misma clasificación ($c(x)$)
- asignar una probabilidad basada en frecuencias observadas en valores de A en nodo n
 - Ej. en 6 ejemplos, $A=1$ y en 4 ejemplos $A=0$, luego $P(A(x) = 1) = 0.6$ y $P(A(x) = 0) = 0.4$

Índice

- Árboles de decisión
- Algoritmo
- Adecuación a valores continuos
- Valores faltantes
- **Atributos con costo**
- Resumen

Atributos con costos

Ej. Estudios médicos

Atributo	Costo	Atributo	Costo
Temperatura	x	ResultadoBiopsia	200 x
Pulso	x	ResultadoLaboratorio	50 x

Distintos costos: económicos y confort del paciente.

- Preferimos árboles que usen atributos de bajo costo usando los de alto costo sólo cuando es necesario.
- **Modificación ID3:** se usa el término de costo en medida selección de atributo
 - $\text{Gain}(S,A) / \text{costo}(A)$ (se prefieren atributos de menor costo)

Índice

- Árboles de decisión
- Algoritmo
- Adecuación a valores continuos
- Valores faltantes
- Atributos con costo
- **Resumen**

ID-3, C4.5, CART...

- Criterio de selección de atributos (splitting criteria)
 - ID-3: Information Gain.
 - C4.5: Gain Ratio
 - CART: Gini
 - CHAID: Chi cuadrado
- Tipo de valores
 - ID-3: Categóricos
 - C4.5 y CART: Categóricos y numéricos
- Valores faltantes (missing values)
 - ID-3 no los trata
 - C4.5, CART los tratan
- Estrategia de poda
 - ID-3: sin poda
 - C4.5: Error-based pruning

Resumen

- aprendizaje supervisado.
- para clasificación y regresión
- fáciles de usar y de entender
- buen método exploratorio para ver qué atributos son importantes
- (tipo de generalización, sesgo, overfitting)

Ventajas:

- fácil visualización e interpretación
- se pueden usar atributos categóricos, continuos, binarios

Desventajas:

- pueden tener sobreajuste
- suelen necesitarse ensambles de árboles para tener mejor performance

Bibliografía

Capítulos de libros:

- .Mitchell, Cap. 3
- .Alpaydin, Cap. 9
- .Marsland, Cap. 12

Artículos:

- . Induction of Decision Trees . Quinlan. <http://hunch.net/~coms-4771/quinlan.pdf>
- . Simplifying Decision Trees. Quinlan.
<https://www.sciencedirect.com/science/article/pii/S0020737387800536>