Clase 4
Aprendizaje de Árboles de Decisión

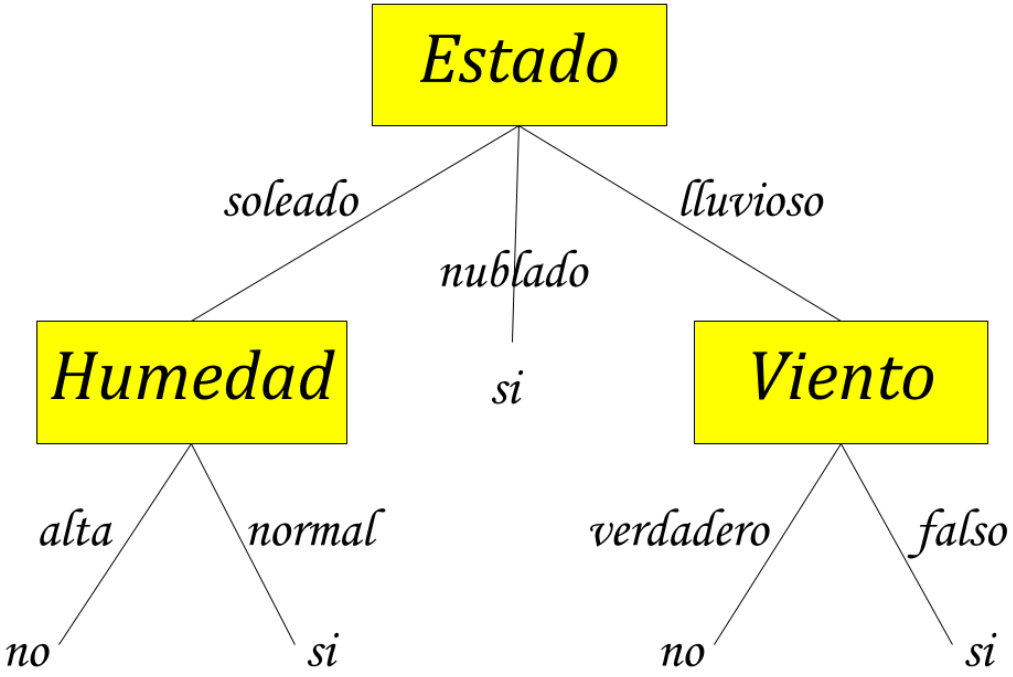
Marcelo Luis Errecalde[1,2]

[1]Universidad Nacional de San Luis, Argentina
[2]Universidad Nacional de la Patagonia Austral, Argentina
e-mails: merreca@unsl.edu.ar, merrecalde@gmail.com

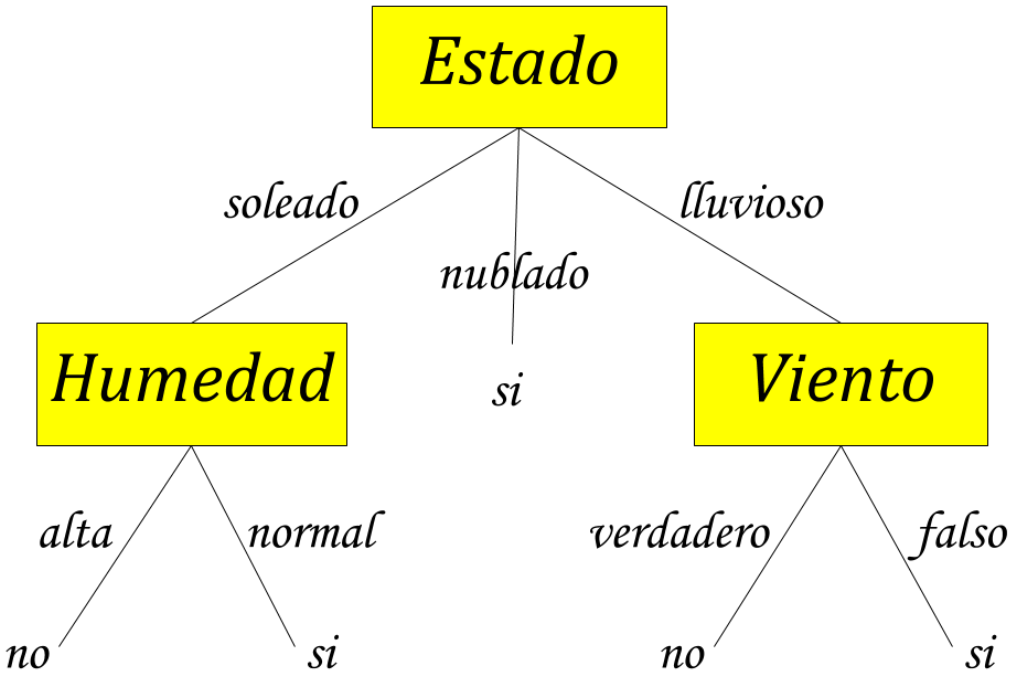
Curso: Minería de Datos
Universidad Nacional de San Luis - Año 2018

Agenda

Árbol de decisión para el concepto *JugarTenis*



Árbol de decisión para el concepto *JugarTenis*



¿Cómo clasificaría este árbol la siguiente instancia?

$\langle Estado = soleado, Temperatura = caluroso, Humedad = alta, Viento = verdadero \rangle$

Representación de árboles de decisión

- Cada nodo interno testea un atributo.
- Cada rama corresponde a un valor del atributo.
- Cada nodo hoja asigna una clasificación.

Los árboles de decisión representan una disyunción de conjunciones de restricciones sobre los valores de atributo de las instancias.

$$(Estado = soleado \ \wedge \ Humedad = normal)$$

$$\vee$$

$$(Estado = Nublado)$$

$$\vee$$

$$(Estado = lluvioso \ \wedge \ Viento = falso)$$

Cuando usar árboles de decisión

- Las instancias son representadas por pares atributo-valor.
- La función objetivo tiene valores de salida discretos.
- Pueden requerirse descripciones disyuntivas.
- Los datos de entrenamiento pueden contener errores.
- Los datos de entrenamiento pueden tener valores de atributos desconocidos.

Ejemplos:

- Diagnóstico médico o de equipamiento.
- Análisis de riesgo de créditos.
- etc.

Aprendizaje de árboles de decisión

función $ID3(E, A_{\mathcal{O}}, Atributos)$

entrada: E, un conjunto de ejemplos de entrenamiento.

$A_{\mathcal{O}}$, el atributo cuyo valor el árbol debe predecir.

$Atributos$, el resto de los atributos que pueden ser testeados por el árbol de decisión.

salida: Un árbol de decisión.

Aprendizaje de árboles de decisión

función $ID3(E, A_{\mathcal{O}}, Atributos)$

1) Crear un nodo *Raíz* para el árbol

2) Si $\forall e \in E, e(A_{\mathcal{O}}) = v_j$, retornar el árbol de un único nodo *Raíz*, con rótulo v_j

3) Si $Atributos = \emptyset$, retornar el árbol de un único nodo *Raíz*, con rótulo $mcm(A_{\mathcal{O}})$

4) $A \leftarrow$ el atributo que mejor clasifica E

5) *Raíz* $\leftarrow A$

6) Por cada valor posible, $v_i \in V(A)$,

6.a) Agregar una nueva rama debajo de *Raíz*, correspondiente al test $A = v_i$

6.b) Sea E_{v_i} el subconjunto de E que tienen valor v_i para A

6.c) Si $E_{v_i} = \emptyset$

6.c.1) entonces debajo de esta nueva rama agregar un nodo hoja con rótulo $mcm(A_{\mathcal{O}})$

6.c.2) sino debajo de esta nueva rama agregar el subárbol dado por

$$ID3(E_{v_i}, A_{\mathcal{O}}, Atributos - \{A\})$$

7) retornar *Raíz*

Aprendizaje de árboles de decisión

función $ID3(E, A_{\mathcal{O}}, Atributos)$

1) Crear un nodo *Raíz* para el árbol

2) Si $\forall e \in E, e(A_{\mathcal{O}}) = v_j$, retornar el árbol de un único nodo *Raíz*, con rótulo v_j

3) Si $Atributos = \emptyset$, retornar el árbol de un único nodo *Raíz*, con rótulo $mcm(A_{\mathcal{O}})$

4) $A \leftarrow$ el atributo que mejor clasifica E

5) *Raíz* $\leftarrow A$

6) Por cada valor posible, $v_i \in V(A)$,

6.a) Agregar una nueva rama debajo de *Raíz*, correspondiente al test $A = v_i$

6.b) Sea E_{v_i} el subconjunto de E que tienen valor v_i para A

6.c) Si $E_{v_i} = \emptyset$

6.c.1) entonces debajo de esta nueva rama agregar un nodo hoja con rótulo $mcm(A_{\mathcal{O}})$

6.c.2) sino debajo de esta nueva rama agregar el subárbol dado por

$$ID3(E_{v_i}, A_{\mathcal{O}}, Atributos - \{A\})$$

7) retornar *Raíz*

Aprendizaje de árboles de decisión

función $ID3(E, A_{\mathcal{O}}, Atributos)$

1) Crear un nodo *Raíz* para el árbol

2) Si $\forall e \in E, e(A_{\mathcal{O}}) = v_j$, retornar el árbol de un único nodo *Raíz*, con rótulo v_j

3) Si $Atributos = \emptyset$, retornar el árbol de un único nodo *Raíz*, con rótulo $mcm(A_{\mathcal{O}})$

4) $A \leftarrow$ el atributo que mejor clasifica E

5) *Raíz* $\leftarrow A$

6) Por cada valor posible, $v_i \in V(A)$,

6.a) Agregar una nueva rama debajo de *Raíz*, correspondiente al test $A = v_i$

6.b) Sea E_{v_i} el subconjunto de E que tienen valor v_i para A

6.c) Si $E_{v_i} = \emptyset$

6.c.1) entonces debajo de esta nueva rama agregar un nodo hoja con rótulo $mcm(A_{\mathcal{O}})$

6.c.2) sino debajo de esta nueva rama agregar el subárbol dado por

$$ID3(E_{v_i}, A_{\mathcal{O}}, Atributos - \{A\})$$

7) retornar *Raíz*

Aprendizaje de árboles de decisión

función $ID3(E, A_{\mathcal{O}}, Atributos)$

1) Crear un nodo *Raíz* para el árbol

2) Si $\forall e \in E$, $e(A_{\mathcal{O}}) = v_j$, retornar el árbol de un único nodo *Raíz*, con rótulo v_j

3) Si $Atributos = \emptyset$, retornar el árbol de un único nodo *Raíz*, con rótulo $mcm(A_{\mathcal{O}})$

4) $A \leftarrow$ el atributo que mejor clasifica E

5) *Raíz* $\leftarrow A$

6) Por cada valor posible, $v_i \in V(A)$,

6.a) Agregar una nueva rama debajo de *Raíz*, correspondiente al test $A = v_i$

6.b) Sea E_{v_i} el subconjunto de E que tienen valor v_i para A

6.c) Si $E_{v_i} = \emptyset$

6.c.1) entonces debajo de esta nueva rama agregar un nodo hoja con rótulo $mcm(A_{\mathcal{O}})$

6.c.2) sino debajo de esta nueva rama agregar el subárbol dado por

$$ID3(E_{v_i}, A_{\mathcal{O}}, Atributos - \{A\})$$

7) retornar *Raíz*

Aprendizaje de árboles de decisión

función $ID3(E, A_{\mathcal{O}}, Atributos)$

1) Crear un nodo *Raíz* para el árbol

2) Si $\forall e \in E, e(A_{\mathcal{O}}) = v_j$, retornar el árbol de un único nodo *Raíz*, con rótulo v_j

3) Si $Atributos = \emptyset$, retornar el árbol de un único nodo *Raíz*, con rótulo $mcm(A_{\mathcal{O}})$

4) $A \leftarrow$ el atributo que mejor clasifica E

5) *Raíz* $\leftarrow A$

6) Por cada valor posible, $v_i \in V(A)$,

6.a) Agregar una nueva rama debajo de *Raíz*, correspondiente al test $A = v_i$

6.b) Sea E_{v_i} el subconjunto de E que tienen valor v_i para A

6.c) Si $E_{v_i} = \emptyset$

6.c.1) entonces debajo de esta nueva rama agregar un nodo hoja con rótulo $mcm(A_{\mathcal{O}})$

6.c.2) sino debajo de esta nueva rama agregar el subárbol dado por

$$ID3(E_{v_i}, A_{\mathcal{O}}, Atributos - \{A\})$$

7) retornar *Raíz*

Aprendizaje de árboles de decisión

función $ID3(E, A_{\mathcal{O}}, Atributos)$

1) Crear un nodo *Raíz* para el árbol

2) Si $\forall e \in E, e(A_{\mathcal{O}}) = v_j$, retornar el árbol de un único nodo *Raíz*, con rótulo v_j

3) Si $Atributos = \emptyset$, retornar el árbol de un único nodo *Raíz*, con rótulo $mcm(A_{\mathcal{O}})$

4) $A \leftarrow$ el atributo que mejor clasifica E

5) *Raíz* $\leftarrow A$

6) Por cada valor posible, $v_i \in V(A)$,

6.a) Agregar una nueva rama debajo de *Raíz*, correspondiente al test $A = v_i$

6.b) Sea E_{v_i} el subconjunto de E que tienen valor v_i para A

6.c) Si $E_{v_i} = \emptyset$

6.c.1) entonces debajo de esta nueva rama agregar un nodo hoja con rótulo $mcm(A_{\mathcal{O}})$

6.c.2) sino debajo de esta nueva rama agregar el subárbol dado por

$$ID3(E_{v_i}, A_{\mathcal{O}}, Atributos - \{A\})$$

7) retornar *Raíz*

Aprendizaje de árboles de decisión

función $ID3(E, A_{\mathcal{O}}, Atributos)$

1) Crear un nodo *Raíz* para el árbol

2) Si $\forall e \in E, e(A_{\mathcal{O}}) = v_j$, retornar el árbol de un único nodo *Raíz*, con rótulo v_j

3) Si $Atributos = \emptyset$, retornar el árbol de un único nodo *Raíz*, con rótulo $mcm(A_{\mathcal{O}})$

4) $A \leftarrow$ el atributo que mejor clasifica E

5) *Raíz* $\leftarrow A$

6) Por cada valor posible, $v_i \in V(A)$,

6.a) Agregar una nueva rama debajo de *Raíz*, correspondiente al test $A = v_i$

6.b) Sea E_{v_i} el subconjunto de E que tienen valor v_i para A

6.c) Si $E_{v_i} = \emptyset$

6.c.1) entonces debajo de esta nueva rama agregar un nodo hoja con rótulo $mcm(A_{\mathcal{O}})$

6.c.2) sino debajo de esta nueva rama agregar el subárbol dado por

$$ID3(E_{v_i}, A_{\mathcal{O}}, Atributos - \{A\})$$

7) retornar *Raíz*

Aprendizaje de árboles de decisión

función $ID3(E, A_{\mathcal{O}}, Atributos)$

1) Crear un nodo *Raíz* para el árbol

2) Si $\forall e \in E, e(A_{\mathcal{O}}) = v_j$, retornar el árbol de un único nodo *Raíz*, con rótulo v_j

3) Si $Atributos = \emptyset$, retornar el árbol de un único nodo *Raíz*, con rótulo $mcm(A_{\mathcal{O}})$

4) $A \leftarrow$ el atributo que mejor clasifica E

5)*Raíz* $\leftarrow A$

6) Por cada valor posible, $v_i \in V(A)$,

6.a) Agregar una nueva rama debajo de *Raíz*, correspondiente al test $A = v_i$

6.b) Sea E_{v_i} el subconjunto de E que tienen valor v_i para A

6.c) Si $E_{v_i} = \emptyset$

6.c.1) entonces debajo de esta nueva rama agregar un nodo hoja con rótulo $mcm(A_{\mathcal{O}})$

6.c.2) sino debajo de esta nueva rama agregar el subárbol dado por

$$ID3(E_{v_i}, A_{\mathcal{O}}, Atributos - \{A\})$$

7) retornar *Raíz*

Aprendizaje de árboles de decisión

función $ID3(E, A_{\mathcal{O}}, Atributos)$

1) Crear un nodo *Raíz* para el árbol

2) Si $\forall e \in E, e(A_{\mathcal{O}}) = v_j$, retornar el árbol de un único nodo *Raíz*, con rótulo v_j

3) Si $Atributos = \emptyset$, retornar el árbol de un único nodo *Raíz*, con rótulo $mcm(A_{\mathcal{O}})$

4) $A \leftarrow$ el atributo que mejor clasifica E

5)*Raíz* $\leftarrow A$

6) Por cada valor posible, $v_i \in V(A)$,

6.a) Agregar una nueva rama debajo de *Raíz*, correspondiente al test $A = v_i$

6.b) Sea E_{v_i} el subconjunto de E que tienen valor v_i para A

6.c) Si $E_{v_i} = \emptyset$

6.c.1) entonces debajo de esta nueva rama agregar un nodo hoja con rótulo $mcm(A_{\mathcal{O}})$

6.c.2) sino debajo de esta nueva rama agregar el subárbol dado por

$$ID3(E_{v_i}, A_{\mathcal{O}}, Atributos - \{A\})$$

7) retornar *Raíz*

Aprendizaje de árboles de decisión

función $ID3(E, A_{\mathcal{O}}, Atributos)$

1) Crear un nodo *Raíz* para el árbol

2) Si $\forall e \in E, e(A_{\mathcal{O}}) = v_j$, retornar el árbol de un único nodo *Raíz*, con rótulo v_j

3) Si $Atributos = \emptyset$, retornar el árbol de un único nodo *Raíz*, con rótulo $mcm(A_{\mathcal{O}})$

4) $A \leftarrow$ el atributo que mejor clasifica E

5)*Raíz* $\leftarrow A$

6) Por cada valor posible, $v_i \in V(A)$,

6.a) Agregar una nueva rama debajo de *Raíz*, correspondiente al test $A = v_i$

6.b) Sea E_{v_i} el subconjunto de E que tienen valor v_i para A

6.c) Si $E_{v_i} = \emptyset$

6.c.1) entonces debajo de esta nueva rama agregar un nodo hoja con rótulo $mcm(A_{\mathcal{O}})$

6.c.2) sino debajo de esta nueva rama agregar el subárbol dado por

$$ID3(E_{v_i}, A_{\mathcal{O}}, Atributos - \{A\})$$

7) retornar *Raíz*

Aprendizaje de árboles de decisión

función $ID3(E, A_{\mathcal{O}}, Atributos)$

1) Crear un nodo *Raíz* para el árbol

2) Si $\forall e \in E, e(A_{\mathcal{O}}) = v_j$, retornar el árbol de un único nodo *Raíz*, con rótulo v_j

3) Si $Atributos = \emptyset$, retornar el árbol de un único nodo *Raíz*, con rótulo $mcm(A_{\mathcal{O}})$

4) $A \leftarrow$ el atributo que mejor clasifica E

5)*Raíz* $\leftarrow A$

6) Por cada valor posible, $v_i \in V(A)$,

6.a) Agregar una nueva rama debajo de *Raíz*, correspondiente al test $A = v_i$

6.b) Sea E_{v_i} el subconjunto de E que tienen valor v_i para A

6.c) Si $E_{v_i} = \emptyset$

6.c.1) entonces debajo de esta nueva rama agregar un nodo hoja con rótulo $mcm(A_{\mathcal{O}})$

6.c.2) sino debajo de esta nueva rama agregar el subárbol dado por

$$ID3(E_{v_i}, A_{\mathcal{O}}, Atributos - \{A\})$$

7) retornar *Raíz*

Aprendizaje de árboles de decisión

función $ID3(E, A_{\mathcal{O}}, Atributos)$

1) Crear un nodo *Raíz* para el árbol

2) Si $\forall e \in E$, $e(A_{\mathcal{O}}) = v_j$, retornar el árbol de un único nodo *Raíz*, con rótulo v_j

3) Si $Atributos = \emptyset$, retornar el árbol de un único nodo *Raíz*, con rótulo $mcm(A_{\mathcal{O}})$

4) $A \leftarrow$ el atributo que mejor clasifica E

5)*Raíz* $\leftarrow A$

6) Por cada valor posible, $v_i \in V(A)$,

6.a) Agregar una nueva rama debajo de *Raíz*, correspondiente al test $A = v_i$

6.b) Sea E_{v_i} el subconjunto de E que tienen valor v_i para A

6.c)Si $E_{v_i} = \emptyset$

6.c.1) entonces debajo de esta nueva rama agregar un nodo hoja con rótulo $mcm(A_{\mathcal{O}})$

6.c.2) sino debajo de esta nueva rama agregar el subárbol dado por

$$ID3(E_{v_i}, A_{\mathcal{O}}, Atributos - \{A\})$$

7) retornar *Raíz*

Aprendizaje de árboles de decisión

función $ID3(E, A_{\mathcal{O}}, Atributos)$

1) Crear un nodo *Raíz* para el árbol

2) Si $\forall e \in E, e(A_{\mathcal{O}}) = v_j$, retornar el árbol de un único nodo *Raíz*, con rótulo v_j

3) Si $Atributos = \emptyset$, retornar el árbol de un único nodo *Raíz*, con rótulo $mcm(A_{\mathcal{O}})$

4) $A \leftarrow$ el atributo que mejor clasifica E

5)*Raíz* $\leftarrow A$

6) Por cada valor posible, $v_i \in V(A)$,

6.a) Agregar una nueva rama debajo de *Raíz*, correspondiente al test $A = v_i$

6.b) Sea E_{v_i} el subconjunto de E que tienen valor v_i para A

6.c)Si $E_{v_i} = \emptyset$

6.c.1) entonces debajo de esta nueva rama agregar un nodo hoja con rótulo $mcm(A_{\mathcal{O}})$

6.c.2) sino debajo de esta nueva rama agregar el subárbol dado por

$$ID3(E_{v_i}, A_{\mathcal{O}}, Atributos - \{A\})$$

7) retornar *Raíz*

Aprendizaje de árboles de decisión

función $ID3(E, A_{\mathcal{O}}, Atributos)$

1) Crear un nodo *Raíz* para el árbol

2) Si $\forall e \in E, e(A_{\mathcal{O}}) = v_j$, retornar el árbol de un único nodo *Raíz*, con rótulo v_j

3) Si $Atributos = \emptyset$, retornar el árbol de un único nodo *Raíz*, con rótulo $mcm(A_{\mathcal{O}})$

4) $A \leftarrow$ el atributo que mejor clasifica E

5)*Raíz* $\leftarrow A$

6) Por cada valor posible, $v_i \in V(A)$,

6.a) Agregar una nueva rama debajo de *Raíz*, correspondiente al test $A = v_i$

6.b) Sea E_{v_i} el subconjunto de E que tienen valor v_i para A

6.c)Si $E_{v_i} = \emptyset$

6.c.1) entonces debajo de esta nueva rama agregar un nodo hoja con rótulo $mcm(A_{\mathcal{O}})$

6.c.2) sino debajo de esta nueva rama agregar el subárbol dado por

$$ID3(E_{v_i}, A_{\mathcal{O}}, Atributos - \{A\})$$

7) retornar *Raíz*

Aprendizaje de árboles de decisión

función $ID3(E, A_{\mathcal{O}}, Atributos)$

1) Crear un nodo *Raíz* para el árbol

2) Si $\forall e \in E, e(A_{\mathcal{O}}) = v_j$, retornar el árbol de un único nodo *Raíz*, con rótulo v_j

3) Si $Atributos = \emptyset$, retornar el árbol de un único nodo *Raíz*, con rótulo $mcm(A_{\mathcal{O}})$

4) $A \leftarrow$ el atributo que mejor clasifica E

5)*Raíz* $\leftarrow A$

6) Por cada valor posible, $v_i \in V(A)$,

6.a) Agregar una nueva rama debajo de *Raíz*, correspondiente al test $A = v_i$

6.b) Sea E_{v_i} el subconjunto de E que tienen valor v_i para A

6.c)Si $E_{v_i} = \emptyset$

6.c.1) entonces debajo de esta nueva rama agregar un nodo hoja con rótulo $mcm(A_{\mathcal{O}})$

6.c.2) sino debajo de esta nueva rama agregar el subárbol dado por

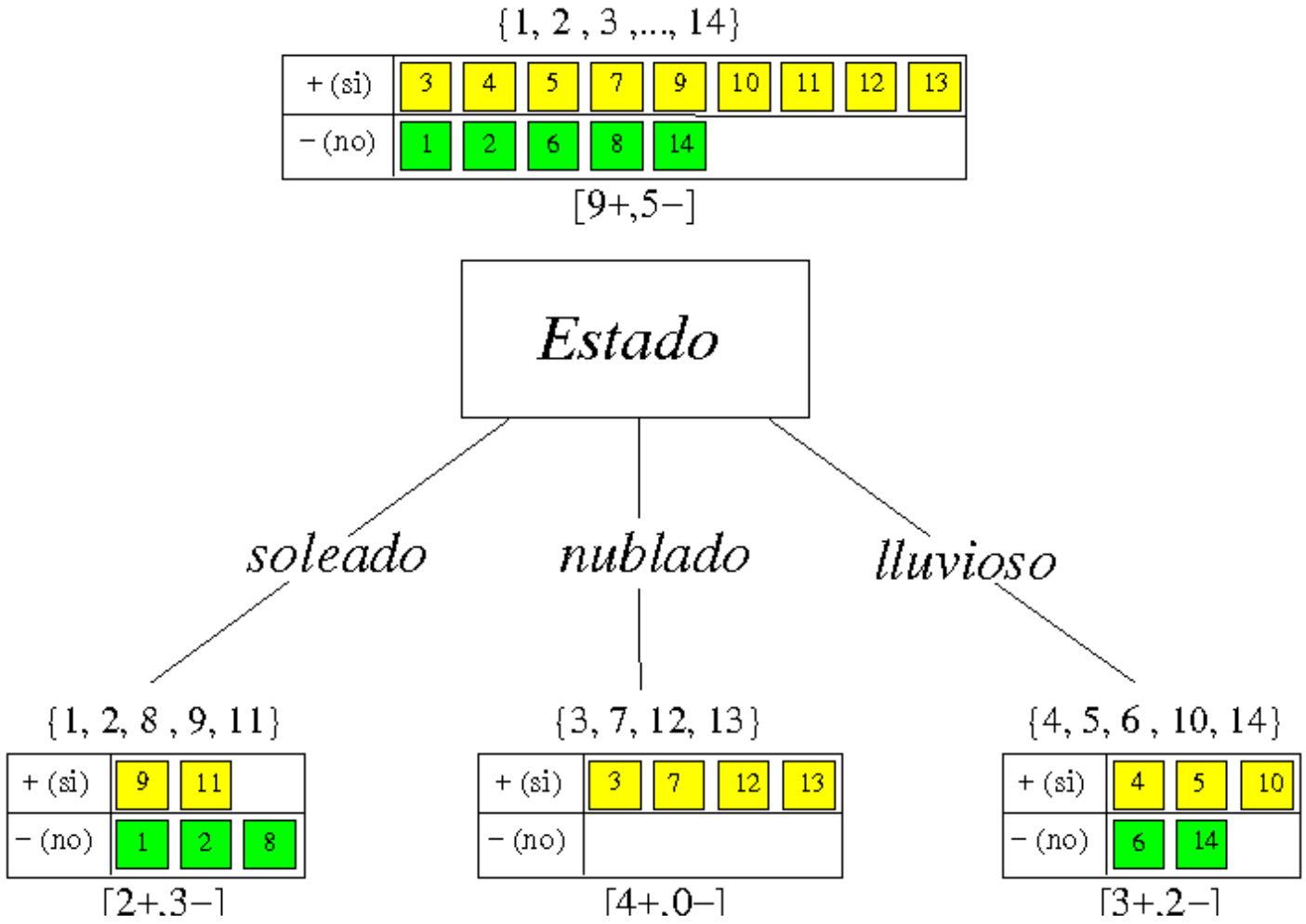
$$ID3(E_{v_i}, A_{\mathcal{O}}, Atributos - \{A\})$$

7) retornar *Raíz*

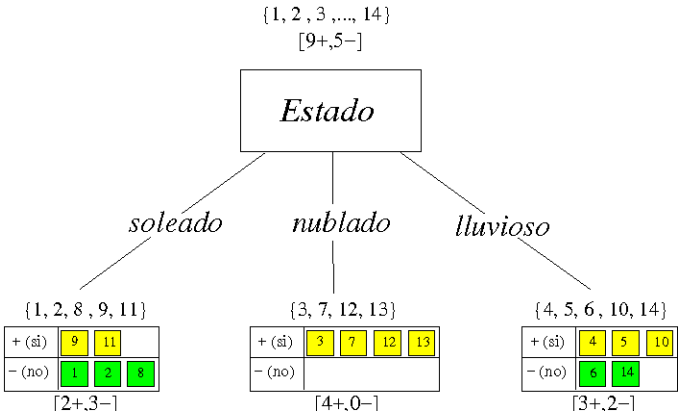
Un conjunto de entrenamiento pequeño

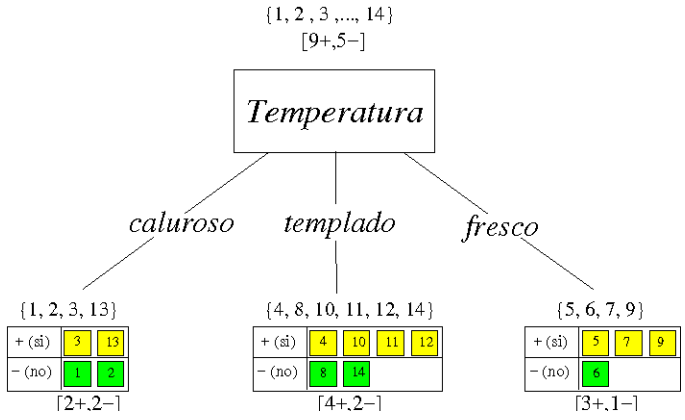
Ejemplo	Atributos				Clase ($A_{\mathcal{O}}$)
	Estado	*Temperatura*	*Humedad*	*Viento*	*JugarTenis*
e_1	*soleado*	*caluroso*	*alta*	*falso*	*no*
e_2	*soleado*	*caluroso*	*alta*	*verdadero*	*no*
e_3	*nublado*	*caluroso*	*alta*	*falso*	*si*
e_4	*lluvioso*	*templado*	*alta*	*falso*	*si*
e_5	*lluvioso*	*fresco*	*normal*	*falso*	*si*
e_6	*lluvioso*	*fresco*	*normal*	*verdadero*	*no*
e_7	*nublado*	*fresco*	*normal*	*verdadero*	*si*
e_8	*soleado*	*templado*	*alta*	*falso*	*no*
e_9	*soleado*	*fresco*	*normal*	*falso*	*si*
e_{10}	*lluvioso*	*templado*	*normal*	*falso*	*si*
e_{11}	*soleado*	*templado*	*normal*	*verdadero*	*si*
e_{12}	*nublado*	*templado*	*alta*	*verdadero*	*si*
e_{13}	*nublado*	*caluroso*	*normal*	*falso*	*si*
e_{14}	*lluvioso*	*templado*	*alta*	*verdadero*	*no*

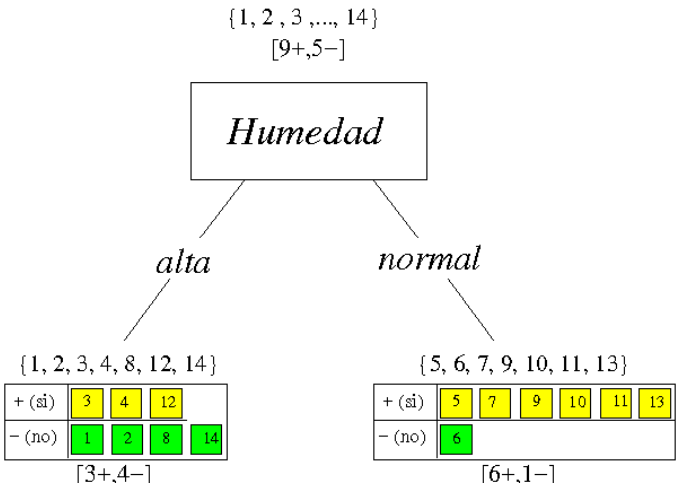
Partición de *E* de acuerdo al atributo *Estado*

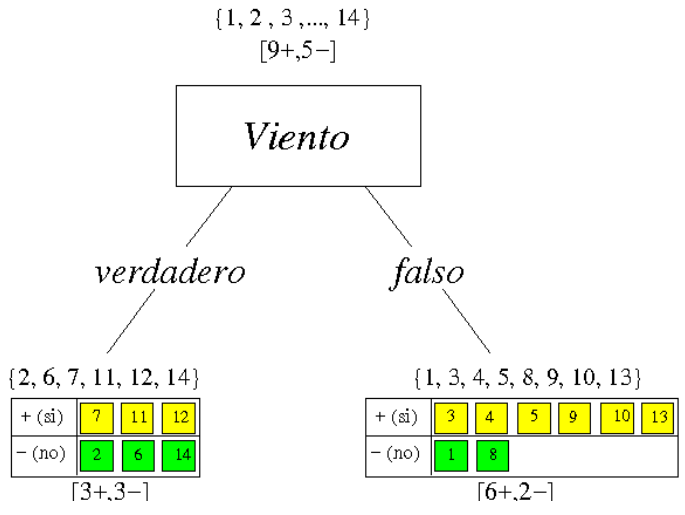


¿Qué atributo elegiría como raíz?









Entropía

La entropía (que denotaremos I) puede ser considerada como la cantidad de información contenida en el resultado de un experimento.

Entropía

La entropía (que denotaremos I) puede ser considerada como la cantidad de información contenida en el resultado de un experimento.

Si un experimento puede tener m resultados distintos $v_1, \ldots, v_m$ que pueden ocurrir con probabilidades $P(v_1), \ldots, P(v_m)$, entonces:

Entropía

La entropía (que denotaremos I) puede ser considerada como la cantidad de información contenida en el resultado de un experimento.

Si un experimento puede tener m resultados distintos $v_1, \ldots, v_m$ que pueden ocurrir con probabilidades $P(v_1), \ldots, P(v_m)$, entonces:

$$I(P(v_1), \ldots, P(v_m)) \equiv \sum_{i=1}^{m} -P(v_i) \log_2 P(v_i)$$

Entropía del conjunto de entrenamiento E

Idea: Considerar a E como una muestra del atributo objetivo $A_{\mathcal{O}}$.

Entropía del conjunto de entrenamiento E

Idea: Considerar a E como una muestra del atributo objetivo $A_{\mathcal{O}}$.

Sea n_{v_k} el número de ejemplos de entrenamiento en E que tienen a v_k como valor de $A_{\mathcal{O}}$. Podemos estimar $P(v_i)$ como la proporción p_{v_i} de ejemplos en E que tienen a v_i como valor de $A_{\mathcal{O}}$:

$$p_{v_i} = \frac{n_{v_i}}{\sum_{v_j \in V(A_{\mathcal{O}})} n_{v_j}}$$

Entropía del conjunto de entrenamiento E

Idea: Considerar a E como una muestra del atributo objetivo $A_{\mathcal{O}}$.

Sea n_{v_k} el número de ejemplos de entrenamiento en E que tienen a v_k como valor de $A_{\mathcal{O}}$. Podemos estimar $P(v_i)$ como la proporción p_{v_i} de ejemplos en E que tienen a v_i como valor de $A_{\mathcal{O}}$:

$$p_{v_i} = \frac{n_{v_i}}{\sum_{v_j \in V(A_{\mathcal{O}})} n_{v_j}}$$

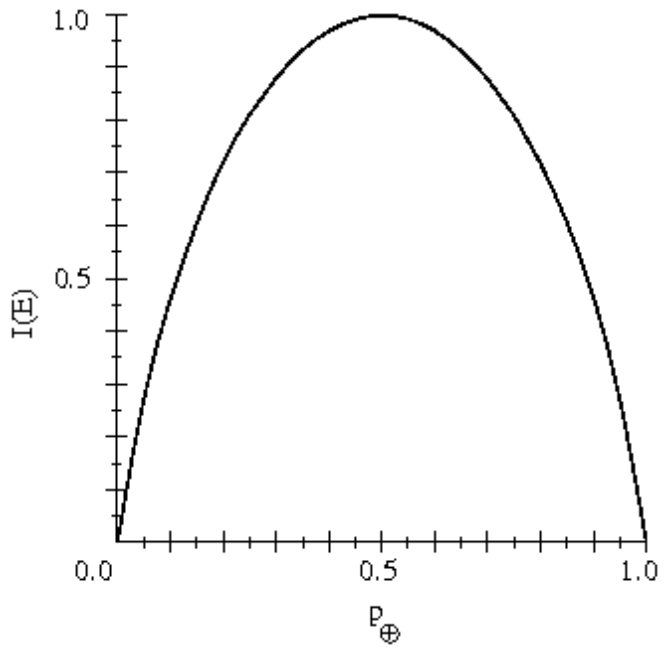
Por lo tanto, la entropía de E respecto a $A_{\mathcal{O}}$ estará dada por:

$$I(E) = I([p_{v_1}, \ldots, p_{v_m}]) = -\sum_{v_j \in V(A_{\mathcal{O}})} p_{v_j} \log_2 p_{v_j}$$

Entropía de E relativa a una clasificación booleana

- $p_\oplus$ es la proporción de ejemplos positivos en E
- $p_\ominus$ es la proporción de ejemplos negativos en E

$$I(E) \equiv -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$$



Información residual de un atributo

Información residual de E respecto a un atributo A ($I_{res}(E, A)$): información que aún necesitaremos para clasificar una instancia después de testear el atributo A.

Información residual de un atributo

Información residual de E respecto a un atributo A ($I_{res}(E, A)$): información que aún necesitaremos para clasificar una instancia después de testear el atributo A.

$$
\begin{aligned}
I_{res}(E, A) &\equiv \sum_{v \in V(A)} P(v) \cdot (- \sum_{c \in V(A_{\mathcal{C}})} P(c|v) \log_2 P(c|v)) \\
&= \sum_{v \in V(A)} P(v) \cdot I(E_v) \\
&= \sum_{v \in V(A)} \frac{|E_v|}{|E|} \cdot I(E_v)
\end{aligned}
$$

Información residual de un atributo

Información residual de E respecto a un atributo A ($I_{res}(E, A)$): información que aún necesitaremos para clasificar una instancia después de testear el atributo A.

$$
\begin{aligned}
I_{res}(E, A) &\equiv \sum_{v \in V(A)} P(v) \cdot (- \sum_{c \in V(A_{\mathcal{O}})} P(c|v) \log_2 P(c|v)) \\
&= \sum_{v \in V(A)} P(v) \cdot I(E_v) \\
&= \sum_{v \in V(A)} \frac{|E_v|}{|E|} \cdot I(E_v)
\end{aligned}
$$

Por lo tanto, la selección de un atributo en el algoritmo de aprendizaje podría limitarse a elegir aquel atributo A con menor información residual $I_{res}(E, A)$.

Ganancia de información de un atributo

Alternativa: considerar la ganancia de información $G(E, A)$ que se obtiene al testear el atributo A.

Ganancia de información de un atributo

Alternativa: considerar la ganancia de información $G(E, A)$ que se obtiene al testear el atributo A. Esta cantidad es la diferencia entre el requerimiento de información original y la información requerida luego de testar el atributo:

$$\begin{aligned} G(E, A) &\equiv I(E) - I_{res}(E, A) \\ &\equiv I(E) - \sum_{v \in V(A)} \frac{|E_v|}{|E|} \cdot I(E_v) \end{aligned}$$

Ganancia de información de un atributo

Alternativa: considerar la ganancia de información $G(E, A)$ que se obtiene al testear el atributo A. Esta cantidad es la diferencia entre el requerimiento de información original y la información requerida luego de testar el atributo:

$$\begin{aligned} G(E, A) &\equiv I(E) - I_{res}(E, A) \\ &\equiv I(E) - \sum_{v \in V(A)} \frac{|E_v|}{|E|} \cdot I(E_v) \end{aligned}$$

Mide la reducción esperada en la entropía al particionar los ejemplos de acuerdo a un atributo.

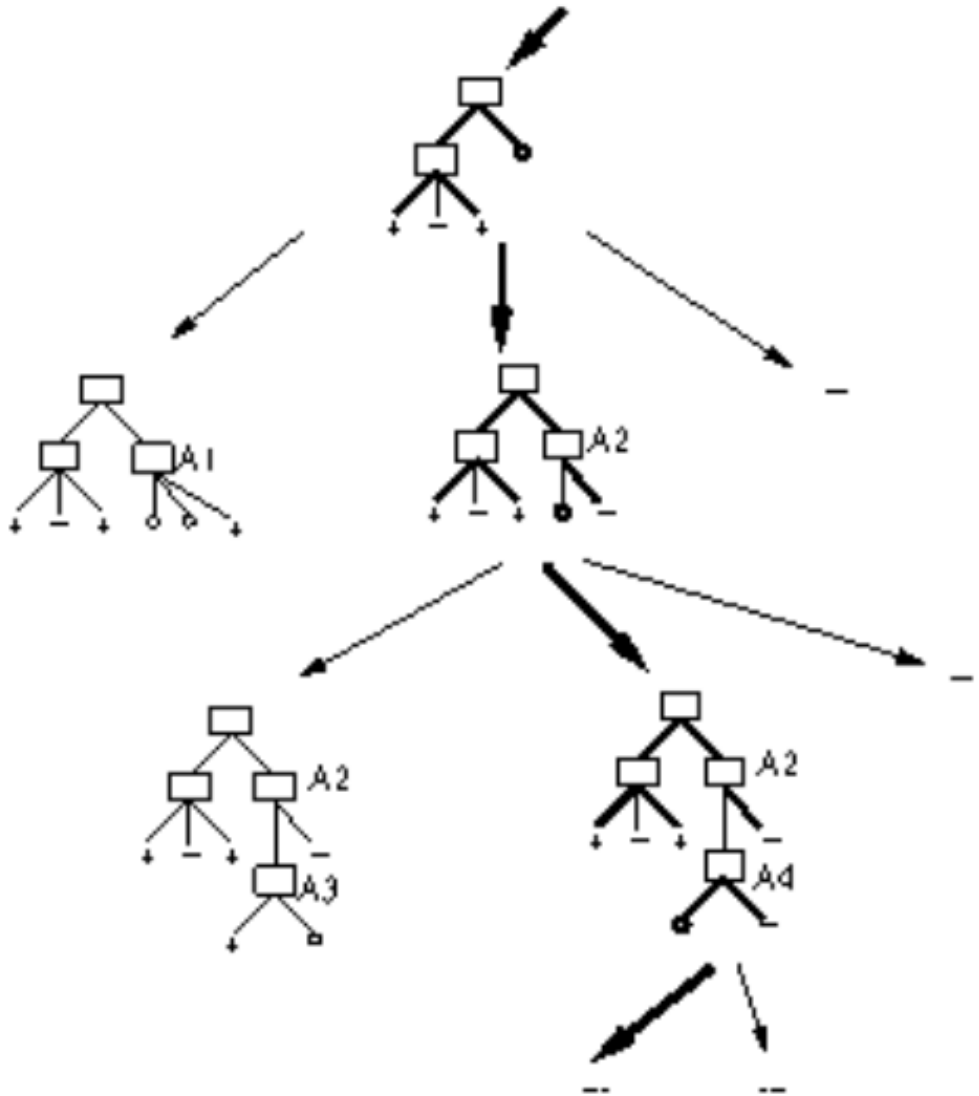
Ganancia de información de un atributo

Alternativa: considerar la ganancia de información $G(E, A)$ que se obtiene al testear el atributo A. Esta cantidad es la diferencia entre el requerimiento de información original y la información requerida luego de testar el atributo:

$$\begin{aligned} G(E, A) &\equiv I(E) - I_{res}(E, A) \\ &\equiv I(E) - \sum_{v \in V(A)} \frac{|E_v|}{|E|} \cdot I(E_v) \end{aligned}$$

Mide la reducción esperada en la entropía al particionar los ejemplos de acuerdo a un atributo. Entonces, se debería elegir aquel con mayor ganancia de información.

Búsqueda en el espacio de hipótesis de ID3



Búsqueda en el espacio de hipótesis de ID3

- El espacio de hipótesis que recorre ID3 es un espacio completo de funciones discretas finitas.

Búsqueda en el espacio de hipótesis de ID3

- El espacio de hipótesis que recorre ID3 es un espacio completo de funciones discretas finitas.
- ID3 mantiene sólo una hipótesis a medida que busca a través del espacio de árboles de decisión.

Búsqueda en el espacio de hipótesis de ID3

- El espacio de hipótesis que recorre ID3 es un espacio completo de funciones discretas finitas.
- ID3 mantiene sólo una hipótesis a medida que busca a través del espacio de árboles de decisión.
- ID3 en su forma pura no realiza backtracking (problema de mínimos locales).

Búsqueda en el espacio de hipótesis de ID3

- El espacio de hipótesis que recorre ID3 es un espacio completo de funciones discretas finitas.
- ID3 mantiene sólo una hipótesis a medida que busca a través del espacio de árboles de decisión.
- ID3 en su forma pura no realiza backtracking (problema de mínimos locales).
- ID3 usa todos los ejemplos de entrenamiento en cada paso de la búsqueda para hacer decisiones basadas en estadísticas (robusto a datos erróneos).

Sesgo inductivo de ID3

Primera aproximación: preferencia por árboles de decisión más cortos sobre árboles complejos.

Sesgo inductivo de ID3

Primera aproximación: preferencia por árboles de decisión más cortos sobre árboles complejos.
Una aproximación más cercana:

> *Los árboles más cortos son preferidos sobre los más largos. A su vez, aquellos árboles que ubican atributos con alta ganancia de información más cerca de la raíz son preferidos a aquellos que no lo hacen.*

Sesgo inductivo de ID3

Primera aproximación: preferencia por árboles de decisión más cortos sobre árboles complejos.
Una aproximación más cercana:

> *Los árboles más cortos son preferidos sobre los más largos. A su vez, aquellos árboles que ubican atributos con alta ganancia de información más cerca de la raíz son preferidos a aquellos que no lo hacen.*

ID3 impone esencialmente un sesgo de búsqueda.

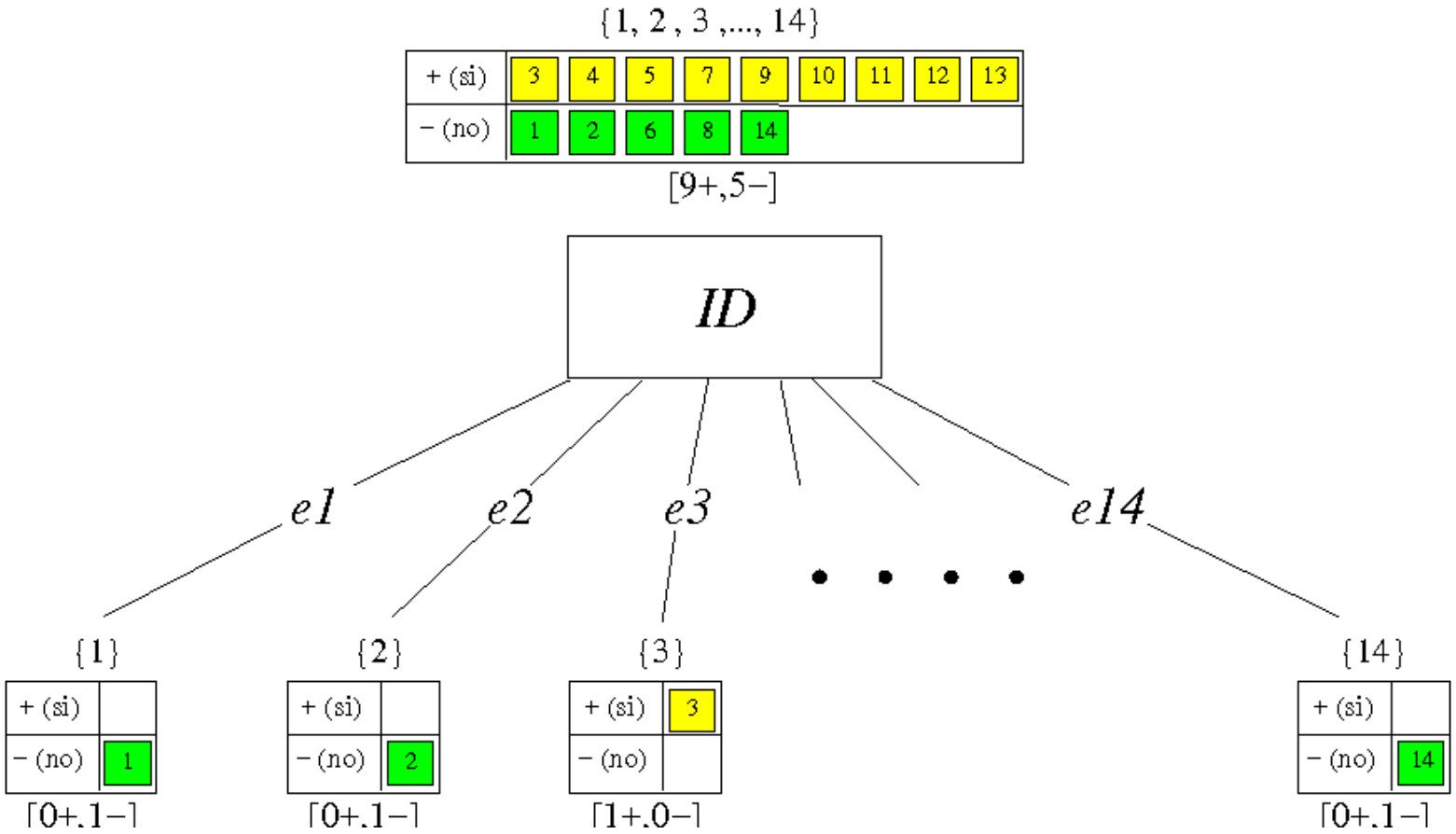
Sesgo inductivo de ID3

Primera aproximación: preferencia por árboles de decisión más cortos sobre árboles complejos.
Una aproximación más cercana:

> *Los árboles más cortos son preferidos sobre los más largos. A su vez, aquellos árboles que ubican atributos con alta ganancia de información más cerca de la raíz son preferidos a aquellos que no lo hacen.*

ID3 impone esencialmente un sesgo de búsqueda.
El sesgo está fundado en el "Occam's razor principle": preferir la hipótesis más simple que se ajusta a los datos.

Problema de atributos con muchos valores



Medidas alternativas para seleccionar atributos

Problema: La ganancia de información tiende a favorecer atributos con muchos valores.

Medidas alternativas para seleccionar atributos

Problema: La ganancia de información tiende a favorecer atributos con muchos valores.
Alternativas:

- Abandonar las medidas basadas en entropía y ganancia de información (usar p. ej. el índice Ginni).
- "Ajustar" la medida de ganancia de información (usar p. ej. el radio de ganancia).

Medidas alternativas para seleccionar atributos

Problema: La ganancia de información tiende a favorecer atributos con muchos valores.
Alternativas:

- Abandonar las medidas basadas en entropía y ganancia de información (usar p. ej. el índice Ginni).
- "Ajustar" la medida de ganancia de información (usar p. ej. el radio de ganancia).

Split Information
$SI(E,A) = -\sum_{v \in V(A)} p_v \log_2 p_v = -\sum_{v \in V(A)} \frac{|E_v|}{|E|} \log_2 \frac{|E_v|}{|E|}$

Medidas alternativas para seleccionar atributos

Problema: La ganancia de información tiende a favorecer atributos con muchos valores.
Alternativas:

- Abandonar las medidas basadas en entropía y ganancia de información (usar p. ej. el índice Ginni).
- "Ajustar" la medida de ganancia de información (usar p. ej. el radio de ganancia).

Split Information
$SI(E,A) = -\sum_{v \in V(A)} p_v \log_2 p_v = -\sum_{v \in V(A)} \frac{|E_v|}{|E|} \log_2 \frac{|E_v|}{|E|}$
Radio de ganancia

$$GR(E,A) \equiv \frac{G(E,A)}{SI(E,A)}$$

Atributos numéricos

Idea: Convertir el atributo numérico A en un atributo booleano A_c. A_c será verdadero si $A < c$ y falso en otro caso.

Atributos numéricos

Idea: Convertir el atributo numérico A en un atributo booleano A_c. A_c será verdadero si $A < c$ y falso en otro caso.

¿Cuáles son los umbrales candidatos?: Aquellos puntos en que se producen los cambios en la clasificación objetivo.

Temperatura:	40	48	60	72	80	90
JugarTenis:	no	no	si	si	si	no

Atributos numéricos

Idea: Convertir el atributo numérico A en un atributo booleano A_c. A_c será verdadero si $A < c$ y falso en otro caso.

¿Cuáles son los umbrales candidatos?: Aquellos puntos en que se producen los cambios en la clasificación objetivo.

Temperatura:	40	48	60	72	80	90
JugarTenis:	no	no	si	si	si	no

Umbrales candidatos: $c = 54$ y $c = 85$. Luego seleccionar el umbral que produce mayor ganancia de información. En este caso, seleccionar entre $Temperatura_{<54}$ y $Temperatura_{<85}$.

Ejemplos con valores desconocidos de los atributos

Situación: Ejemplo de entrenamiento $e = \langle x, c(x) \rangle \in E$ tiene un valor desconocido para el atributo A.

Ejemplos con valores desconocidos de los atributos

Situación: Ejemplo de entrenamiento $e = \langle x, c(x) \rangle \in E$ tiene un valor desconocido para el atributo A. ¿Cómo estimamos $x(A)$ para poder calcular $G(E, A)$ en el nodo n?

Ejemplos con valores desconocidos de los atributos

Situación: Ejemplo de entrenamiento $e = \langle x, c(x) \rangle \in E$ tiene un valor desconocido para el atributo A. ¿Cómo estimamos $x(A)$ para poder calcular $G(E, A)$ en el nodo n?

Alternativas:

- $x(A)$ es el valor más común del atributo A entre los ejemplos de E.

Ejemplos con valores desconocidos de los atributos

Situación: Ejemplo de entrenamiento $e = \langle x, c(x) \rangle \in E$ tiene un valor desconocido para el atributo A. ¿Cómo estimamos $x(A)$ para poder calcular $G(E, A)$ en el nodo n?

Alternativas:

- $x(A)$ es el valor más común del atributo A entre los ejemplos de E.
- $x(A)$ es el valor más común del atributo A entre los ejemplos de E que tienen la clasificación $c(x)$.

Ejemplos con valores desconocidos de los atributos

Situación: Ejemplo de entrenamiento $e = \langle x, c(x) \rangle \in E$ tiene un valor desconocido para el atributo A. ¿Cómo estimamos $x(A)$ para poder calcular $G(E, A)$ en el nodo n?

Alternativas:

- $x(A)$ es el valor más común del atributo A entre los ejemplos de E.
- $x(A)$ es el valor más común del atributo A entre los ejemplos de E que tienen la clasificación $c(x)$.
- Realizar los siguientes pasos:
  - Asignar probabilidad p_i a cada valor posible $v_i \in V(A)$.
  - Calcular la ganancia asignando una "fracción" p_i del ejemplo a cada descendiente en el árbol.

Atributos con diferentes costos

Situación: Algunos atributos son más caros que otros.

Atributos con diferentes costos

Situación: Algunos atributos son más caros que otros.
Idea: Tratar de usar los atributos más baratos y recurrir a los caros sólo cuando se necesitan para producir predicciones confiables.

Atributos con diferentes costos

Situación: Algunos atributos son más caros que otros.
Idea: Tratar de usar los atributos más baratos y recurrir a los caros sólo cuando se necesitan para producir predicciones confiables.
Alternativas. Reemplazar la ganancia por:

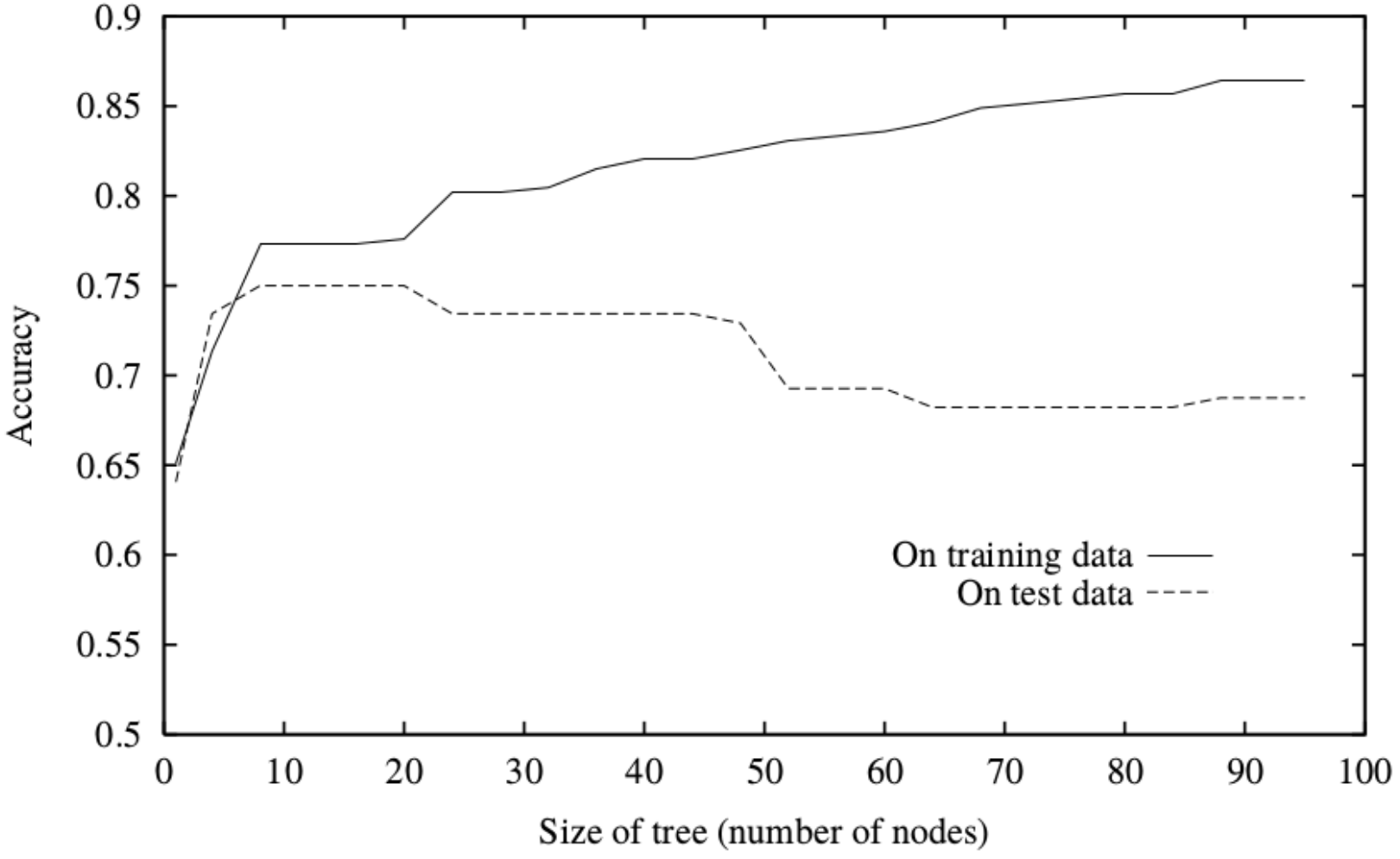
- $\frac{G(E,A)}{Costo(A)}$
- $\frac{G^2(E,A)}{Costo(A)}$ (Tan y Schlimer)
- $\frac{2^{G(E,A)}-1}{(Costo(A)+1)^w}$ (Nunez) donde $w \in [0,1]$ determina la importancia relativa del costo versus la ganancia de información.

El problema del *sobreajuste (overfitting)*

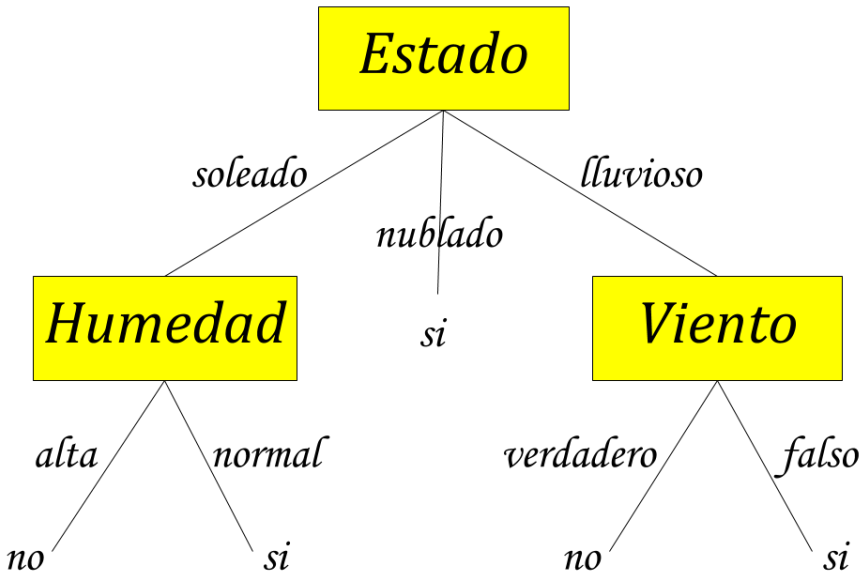
Cuando hay ruido en los datos de entrenamiento o éstos son insuficientes, el algoritmo de aprendizaje puede producir árboles que sobreajustan los ejemplos de entrenamiento.

Definición: Dado un espacio de hipótesis H, una hipótesis $h \in H$ se dice que sobreajusta los datos de entrenamiento si existe alguna hipótesis alternativa $h' \in H$, tal que h tiene un error más pequeño que h' sobre los ejemplos de entrenamiento, pero h' tiene un error más pequeño que h sobre la distribución completa de instancias.

Ejemplo del impacto del *sobreajuste*



Ejemplo de sobreajuste por datos con errores (ruido)



¿Qué sucede si agregamos al conjunto E el siguiente ejemplo positivo, incorrectamente clasificado?

$\langle Estado = soleado, Temperatura = caluroso, Humedad = normal, Viento = verdadero, JugarTenis = no \rangle$

Enfoques para evitar el sobreajuste en ID3

Idea: modificar los algoritmos para obtener modelos más generales.
La generalización se logra eliminando condiciones de la rama del árbol (podando).
Alternativas:

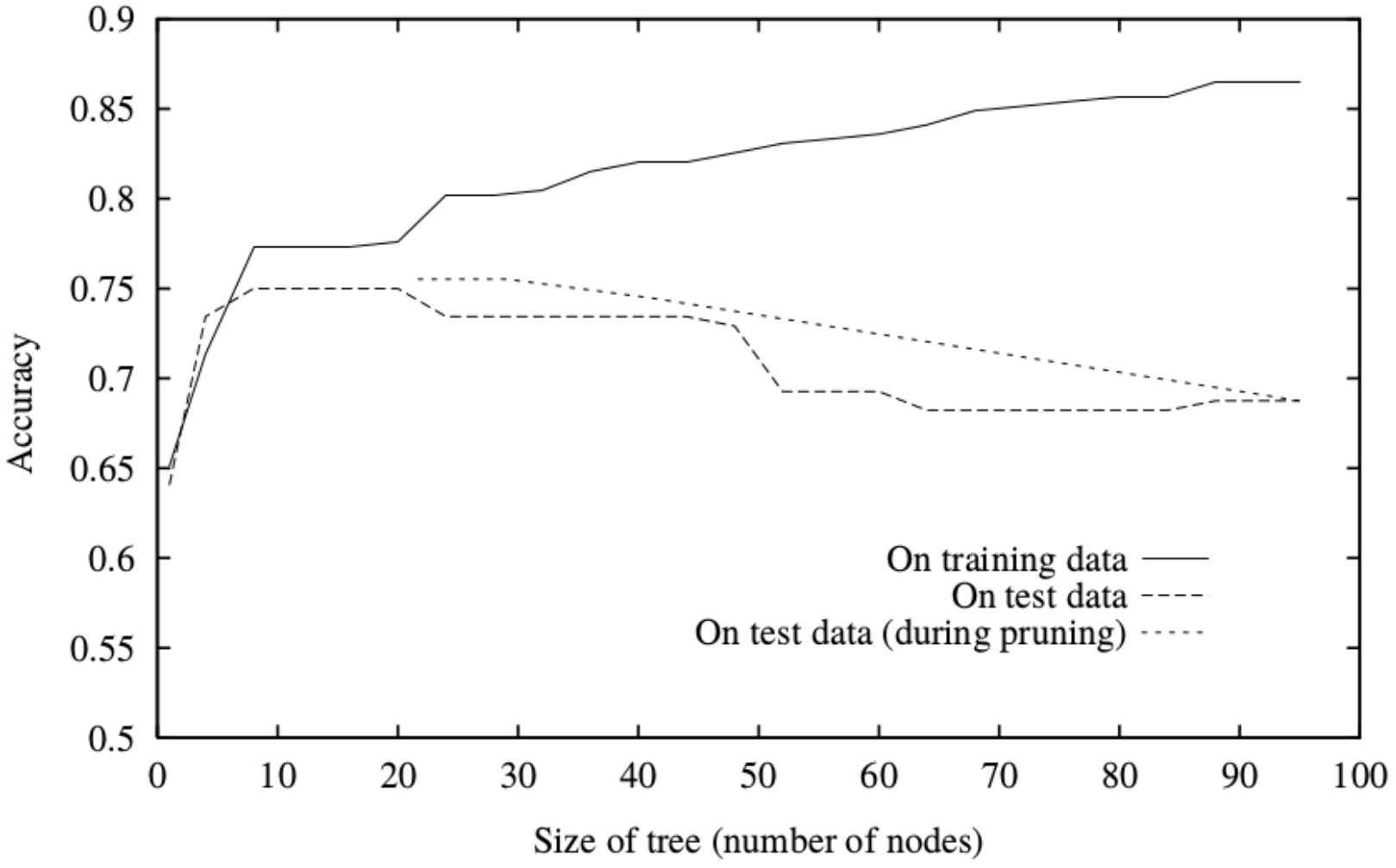
- Mediante prepoda: enfoques que detienen anticipadamente el crecimiento del árbol. El punto central consiste en determinar el criterio de parada a la hora de especializar una rama (por ejemplo, número de ejemplos por nodo).

Enfoques para evitar el sobreajuste en ID3

Idea: modificar los algoritmos para obtener modelos más generales.
La generalización se logra eliminando condiciones de la rama del árbol (podando).
Alternativas:

- Mediante pospoda: enfoques que permiten que el árbol sobreajuste los datos y luego lo podan. Se utiliza generalmente un conjunto de validación. Un ejemplo clásico de poda consiste en eliminar nodos de abajo hacia arriba hasta un cierto límite.

Poda usando un conjunto de validación



¿Preguntas?