Inteligencia Artificial
Hoja 3: Aprendizaje Automático

Publicación: 2012/04/16
Entrega: 2012/04/23/23:55
Resolución en clase: 2012/04/24

Escuela Politécnica Superior

1. *En medicina, se denomina prevalencia de una enfermedad a la probabilidad prior de que un sujeto esté enfermo. Se denomina sensibilidad de una prueba o test a la probabilidad de que dicho test resulte positivo cuando el sujeto está realmente enfermo. Se llama especificidad a la probabilidad de que el test resulte negativo para un paciente sano.*
Considera una enfermedad con una prevalencia de 1 por cada 100 habitantes y un test para diagnosticarla con una sensibilidad del 95% y una especificidad del 90%.

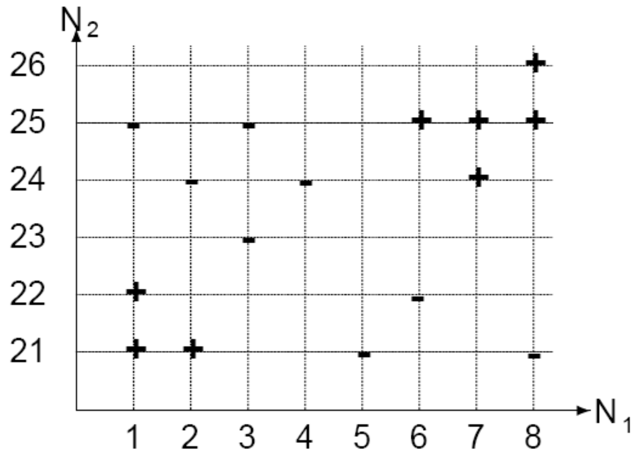
- *¿Cuál es la probabilidad de que un paciente que ha dado positivo en el test esté en realidad sano (falso positivo)?*
- *Si un sujeto ha dado positivo para el test diagnóstico y nos piden maximizar la verosimilitud, ¿qué diagnóstico deberíamos emitir?*
- *¿Y si nos piden maximizar la probabilidad a posteriori?*

2. *Considera el siguiente cuadro que representa las respuestas de 9 pacientes a 4 preguntas (A, B, C y D) cuyas posibles respuestas son sí/no. La última columna representa si el paciente ha desarrollado osteoporosis posteriormente.*

Paciente	A	B	C	D	Desarrollo posterior de osteoporosis
1	Sí	Sí	No	Sí	Sí
2	No	Sí	Sí	No	Sí
3	No	No	No	Sí	No
4	Sí	Sí	No	Sí	Sí
5	Sí	No	Sí	Sí	No
6	No	Sí	No	No	Sí
7	Sí	Sí	Sí	No	No
8	Sí	No	Sí	Sí	No
9	Sí	Sí	Sí	Sí	Sí

- *Según el método de Bayes, ¿cuál es la probabilidad de desarrollo de osteoporosis de un paciente que responde afirmativamente a las preguntas A y C?*
- *¿Y según el método Naïve Bayes?*
- *¿Cómo clasificaría el método K-NN un paciente que responda A=No, B=No, C=Sí, D=Sí considerando k=3? ¿Y k=5? Para resolver esta cuestión, considera que se utiliza, para todos los atributos, la codificación Sí=1 y No=0*

3. *Considera el siguiente problema de clasificación, donde tenemos dos clases (+, -) y los atributos numéricos N_1 y N_2:*



Queremos desarrollar un modelo de clasificación con el algoritmo C4.5. Sabiendo que la pregunta que selecciona dicho algoritmo para el nodo raíz es $N_1>6$, ¿cuál será el árbol final que se generará sin aplicar poda?

4. *Considera una neurona artificial con función de activación escalón caracterizada por los parámetros $w_1 = 1.2$, $w_2 = -0.7$ $\theta=0.3$.*
Clasifica mediante esta neurona los siguientes ejemplos:

$$\mathbf{x}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mathbf{x}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \mathbf{x}_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \mathbf{x}_4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

5. *Aplica regresión logística al problema AND (las entradas son x1, x2, que pueden adquirir los valores 0 y 1; la clase 1 corresponde a la salida 1 del AND). Utiliza aprendizaje online y la constante de aprendizaje $\eta = 1$. Supón que los pesos tienen un valor inicial w0 = -0.3, w1 = 0, w2 = 0.4. Muestra los resultados del aprendizaje al cabo de la primera época, suponiendo que los patrones se muestran en el orden en el que aparecen en la tabla.*

$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	t	w_0	w_1	w_2	h	h-t	Δw_0	Δw_1	Δw_2
1	0	0	0	-0.3	0	0.4	0.43	0.43	-0.43	0	0
1	1	0	0	-0.73	0	0.4	0.33	0.33	-0.33	-0.33	0
1	0	1	0	-1.05	-0.33	0.4	…				
1	1	1	1								

1. Solución:

Notación: H = enfermo.

- P(H) = 1/100 = 0.01
P(positivo | H) = 0.95
P(¬positivo | ¬H) = 0.90

P(H | positivo) = 0.95*0.01 / (0.95*0.01 + (1-0.90)*(1-0.01)) = 0.09
P(¬H | positivo) = 1 – P(H | positivo) = 0.81

- P(positivo|H) = 0.95.
P(positivo|¬H) = 1-P(¬positivo | ¬H) = 1-0.9 = 0.1
ML elige entonces el diagnóstico H (es decir, enfermo)

- Como vimos en la primera respuesta, P(H | positivo) = 0.09.
Por tanto, P(¬H | positivo) = 1-0.09 = 0.91.
Luego MAP elige el dignóstico ¬H (es decir, no enfermo)

2. Solución:

- P(osteoporosis | A^C) = #(ost ^A^C) / (# A^C) = 1/4 = 0.25
- Según el método de Naïve Bayes,
P(A^C^ost) ~ P(A | ost)*P(C | ost) * P(ost) =
[#(A^ost) / #ost] * [#(C^ost) / #ost] * [(#ost) / N] = 3 * 2 / 5 / 9= 0.1333

P(A^C^¬ost) ~ P(A | ¬ost)*P(C | ¬ost) * P(¬ost) =
[#(A^ ¬ost) / #¬ost]·[#(C^ ¬ost) / #¬ost]·[(#¬ost)/N] = 3·3 / 4 / 9 = 0.25

Luego según Naïve Bayes, P(A^C | ost) ~ 0.1333/(0.1333 + 0.25) = 0.35
- Xtest: (0, 0, 1, 1)

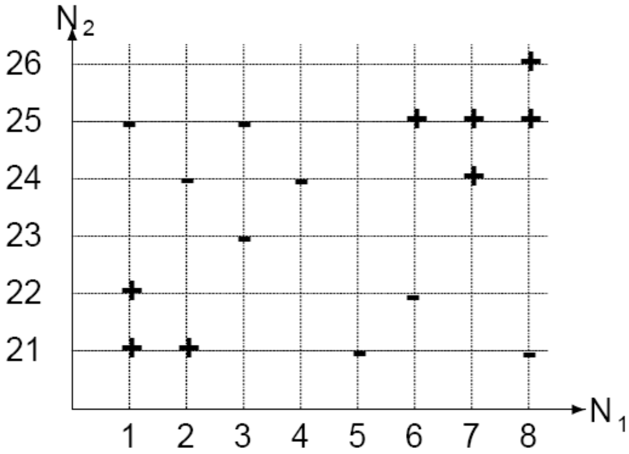
Paciente	A	B	C	D	Clase	distancia^2 con Xtest
1	1	1	0	1	Sí	3
2	0	1	1	0	Sí	2
3	0	0	0	1	No	1
4	1	1	0	1	Sí	3
5	1	0	1	1	No	1
6	0	1	0	0	Sí	3
7	1	1	1	0	No	3
8	1	0	1	1	No	1
9	1	1	1	1	Sí	2
Xtest	0	0	1	1		

Los 3 vecinos más próximos son los pacientes 3, 5 y 8 (los 3 con clase No), con lo cual se predice No.
Los 5 vecinos más próximos son los pacientes 3, 5, 8, 2 y 9 (3 No, 2 Sí), con lo cual se predice No.

3. Solución:

Aunque en el enunciado se daba la primera pregunta que realiza el árbol y no era necesario demostrar que $N_1>6$ es la pregunta que C4.5 realiza en el primer nodo, se incluye aquí esta demostración con fines didácticos:

¿Qué árbol desarrollaría C4.5 con los siguientes datos?



Solución:

Primero vamos a ver qué pregunta elegiría para la raíz.

La entropía de la clase es, ya que hay 8+ y 8-, H(8/16, 8/16) = 1 bits.

Las posibles preguntas con N_1 son:

Pregunta	Rama "No"	Rama "Sí"	Entropía clase en Rama "No"	Entropía clase en Rama "Sí"	H(clase | Pregunta)	IG
$N_1>1$	2+, 1-	6+, 7-	H(2/3, 1/3) = 0.918 bits	H(6/13, 7/13) = 0.996 bits	3/16*0.918+ 13/16*0.996 = 0.981 bits	1-0.981= 0.019 bits
$N_1>2$	3+, 2-	5+, 6-	H(3/5, 2/5) = 0.971 bits	H(5/11, 6/11) = 0.994 bits	5/16*0.971+ 11/16*0.994 = 0.987 bits	1-0.987= 0.013 bits
$N_1>3$	3+, 4-	5+, 4-	H(3/7, 4/7) = 0.985 bits	H(5/9, 4/9) = 0.991 bits	7/16*0.985+ 9/16*0.991 = 0.988 bits	1-0.988= 0.012 bits
$N_1>4$	3+, 5-	5+, 3-	H(3/8, 5/8) = 0.954 bits	H(5/8, 3/8) = 0.954 bits	8/16*0.954+ 8/16*0.954 = 0.954 bits	1-0.954= 0.046 bits
$N_1>5$	3+, 6-	5+, 2-	H(3/9, 6/9) = 0.918 bits	H(5/7, 2/7) = 0.863 bits	9/16*0.918+ 7/16*0.863 = 0.894 bits	1-0.894= 0.106 bits
$N_1>6$	4+, 7-	4+, 1-	H(4/11, 7/11)= 0.946 bits	H(4/5, 1/5) = 0.722 bits	11/16*0.946+ 5/16*0.722 = 0.876 bits	1-0.876= 0.124 bits
$N_1>7$	6+, 7-	2+, 1-	H(6/13, 7/13)= 0.996 bits	H(2/3, 1/3) = 0.918 bits	13/16*0.996+ 3/16*0.918 = 0.981 bits	1-0.981= 0.019 bits

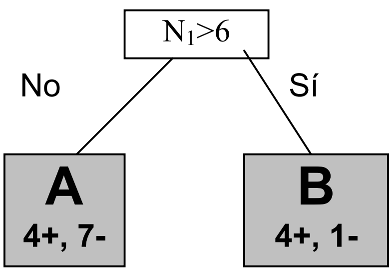
N_1>8	8+, 8-	0+, 0-	1 bit	--	16/16*1+ 0/16*-- = 0 bits	1-1 = 0 bits

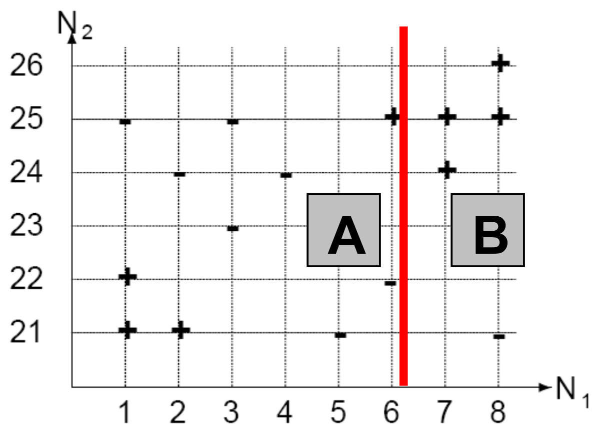
En verde he marcado la pregunta sobre N_1 que ofrece una mayor ganancia de información. Por otra parte, como se puede ver en la tabla, la pregunta sobre el valor más alto del atributo (N_1>6) clasifica a todos los ejemplos por la rama del "No", y a ninguno por la rama del "Sí". Es decir, no hace nada (IG=0). Por tanto siempre podemos evitar chequear esta pregunta en C4.5.

Si ahora estudiamos las posibles preguntas con N_2 (la pregunta sobre el valor más alto, 26, no la chequeamos directamente por lo dicho en el anterior párrafo):

Pregunta	Rama "No"	Rama "Sí"	Entropía clase en Rama "No"	Entropía clase en Rama "Sí"	H(clase | Pregunta)	IG
N_2>21	2+, 2-	6+, 6-	H(2/4, 2/4) = 1 bits	H(6/12, 6/12) = 1 bits	4/16*1+ 12/16*1 = 1 bits	1-1= 0 bits
N_2>22	3+, 3-	5+, 5-	H(3/6, 3/6) = 1 bits	H(5/10, 5/10) = 1 bits	6/16*1+ 10/16*1 = 1 bits	1-1= 0 bits
N_2>23	3+, 4-	5+, 4-	H(3/7, 4/7) = 0.985 bits	H(5/9, 4/9) = 0.991 bits	7/16*0.985+ 9/16*0.991 = 0.988 bits	1-0.988= 0.012 bits
N_2>24	4+, 6-	4+, 2-	H(4/10, 6/10)= 0.971 bits	H(4/6, 2/6) = 0.918 bits	10/16*0.971+ 6/16*0.918 = 0.951 bits	1-0.951= 0.049 bits
N_2>25	7+, 8-	1+, 0-	H(7/15, 8/15)= 0.997 bits	H(1/1, 0/1) = 0 bits	15/16*0.997+ 1/16*0 = 0.935 bits	1-0.894= 0.065 bits

Como vemos, la mejor pregunta que podemos hacer con **N_2** es N_2>25.
Sin embargo, el IG de N_2>25 es menor que el de la mejor pregunta sobre N_1 (N_1>6) así que esta es la pregunta que escogeremos para el nodo raíz.





Expansión del árbol en A

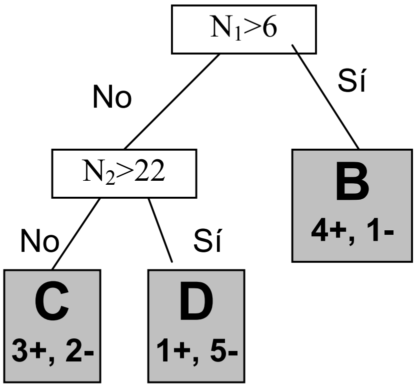
Vemos que hay 4+ y 7- ejemplos, 11 en total.
Primero calculamos H(clase) en **A**, es decir, H(4/11, 7/11) = 0.946 bits.
Ahora analizamos las posibles preguntas sobre N_1:

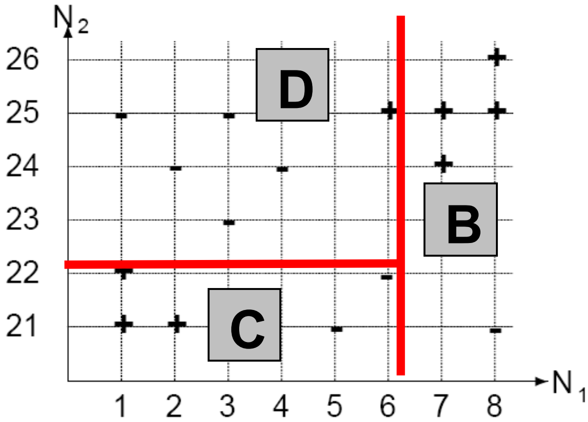
Pregunta	**Rama "No"**	**Rama "Sí"**	**Entropía clase en Rama "No"**	**Entropía clase en Rama "Sí"**	**H(clase | Pre-gunta)**	**IG**
N_1>1	2+, 1-	2+, 6-	H(2/3, 1/3) = 0.918 bits	H(2/8, 6/8) = 0.811 bits	3/11*0.918+ 8/11*0.811 = 0.840 bits	0.946-0.840 = 0.106 bits
N_1>2	3+, 2-	1+, 5-	H(3/5, 2/5) = 0.971 bits	H(1/6, 5/6) = 0.650 bits	5/11*0.971+ 6/11*0.650 = 0.796 bits	0.946-0.796 = 0.150 bits
N_1>3	3+, 4-	1+, 3-	H(3/7, 4/7) = 0.985 bits	H(1/4, 3/4) = 0.811 bits	7/11*0.985+ 4/11*0.811 = 0.922 bits	0.946-0.922 = 0.024 bits
N_1>4	3+, 5-	1+, 2-	H(3/8, 5/8) = 0.954 bits	H(1/3, 2/3) = 0.918 bits	8/11*0.954+ 3/11*0.918 = 0.944 bits	0.946-0.944 = 0.002 bits
N_1>5	3+, 6-	1+, 1-	H(3/9, 6/9) = 0.918 bits	H(1/2, 1/2) = 1 bits	9/11*0.918+ 2/11*1 = 0.933 bits	0.946-0.933 = 0.013 bits

En verde he marcado la pregunta sobre N_1 que ofrece una mayor ganancia de información. Si ahora hacemos lo mismo para N_2:

Pregunta	**Rama "No"**	**Rama "Sí"**	**Entropía clase en Rama "No"**	**Entropía clase en Rama "Sí"**	**H(clase | Pre-gunta)**	**IG**
N_2>21	2+, 1-	2+, 6-	H(2/3, 1/3) = 0.918 bits	H(2/8, 6/8) = 0.811 bits	3/11*0.918+ 8/11*0.811 = 0.840 bits	0.946- 0.840= 0.106 bits
N_2>22	3+, 2-	1+, 5-	H(3/5, 2/5) = 0.971 bits	H(1/6, 5/6) = 0.650 bits	5/11*0.971+ 6/11*0.650 = 0.796 bits	0.946- 0.796= 0.150 bits
N_2>23	3+, 3-	1+, 4-	H(3/6, 3/6) = 1 bits	H(1/5, 4/5) = 0.722 bits	6/11*1+ 5/11*0.722 = 0.874 bits	0.946- 0.874= 0.072 bits
N_2>24	3+, 5-	1+, 2-	H(3/8, 5/8)= 0.954 bits	H(1/3, 2/3) = 0.918 bits	8/11*0.954+ 3/11*0.918 = 0.944 bits	0.946- 0.944= 0.002 bits

Como vemos, la mejor pregunta sobre N_2 "empata" con la mejor pregunta sobre N_1 (tienen exactamente la misma IG). Aunque los valores los hemos redondeado a 3 cifras decimales, las IGs son exactamente las mismas (ambas preguntas separan a los patrones en un subconjunto con 3+, 2- ejemplos y otro subconjunto con 1+, 5-). Vamos a suponer que en este caso de empate C4.5 elige N_2 (podría haber elegido N_1, las dos preguntas son equivalentes en términos de información). Entonces:





Aunque es fácil ver qué va a pasar a continuación, vamos a hacer los cálculos.

Expansión del árbol en **C**

Vemos que en **C** hay 3+ y 2- ejemplos, 5 en total.
Primero calculamos H(clase) en **C**, es decir, H(3/5, 2/5) = 0.971 bits.

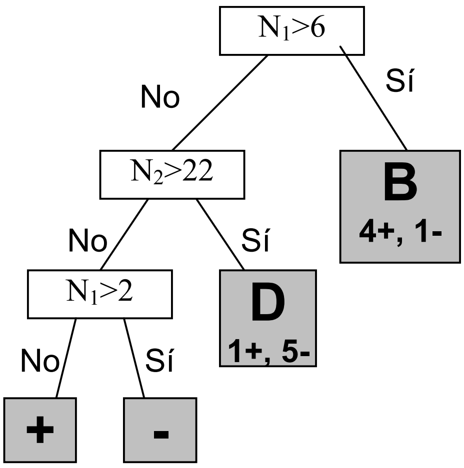
Posibles preguntas sobre N_1:

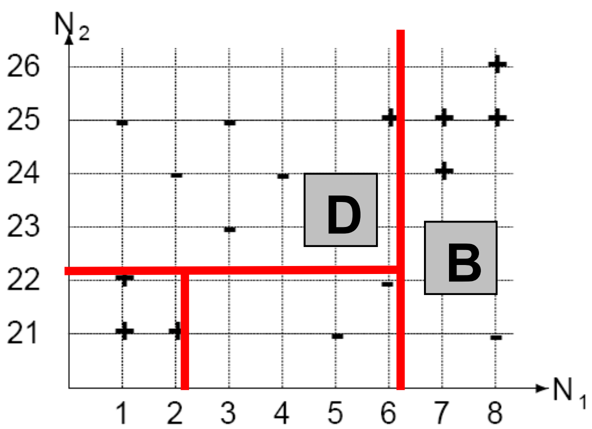
Pregunta	**Rama "No"**	**Rama "Sí"**	**Entropía clase en Rama "No"**	**Entropía clase en Rama "Sí"**	**H(clase | Pregunta)**	**IG**
N_1>1	2+, 0-	1+, 2-	H(2/2, 0/2) = 0 bits	H(1/3, 2/3) = 0.918 bits	2/5*0+ 3/5*0.918 = 0.551 bits	0.971-0.551 = 0.420 bits
N_1>2	3+, 0-	0+, 2-	H(3/3, 0/3) = 0 bits	H(0/2, 2/2) = 0 bits	3/5*0+ 2/5*0 = 0 bits	0.971-0 = 0.971 bits
N_1>5	3+, 1-	0+, 1-	H(3/4, 1/4) = 0.811 bits	H(0/1, 1/1) = 0 bits	4/5*0.811+ 1/5*0 = 0.649 bits	0.971-0.649 = 0.322 bits

En verde he marcado la pregunta sobre N_1 que ofrece una mayor ganancia de información. Si ahora hacemos lo mismo para N_2:

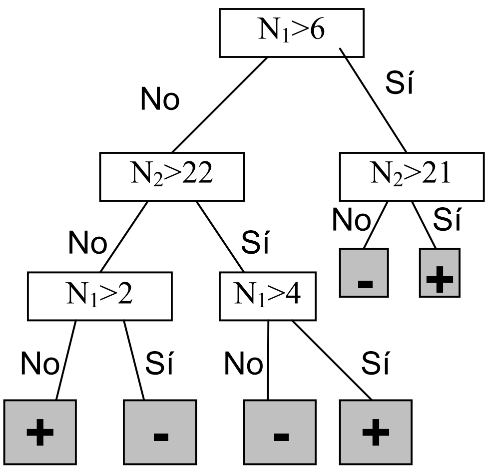
Pregunta	**Rama "No"**	**Rama "Sí"**	**Entropía clase en Rama "No"**	**Entropía clase en Rama "Sí"**	**H(clase | Pregunta)**	**IG**
N_2>21	2+, 1-	1+, 1-	H(2/3, 1/3) = 0.918 bits	H(1/2, 1/2) = 1 bits	3/5*0.918+ 2/5*1 = 0.951 bits	0.971- 0.951= 0.020 bits

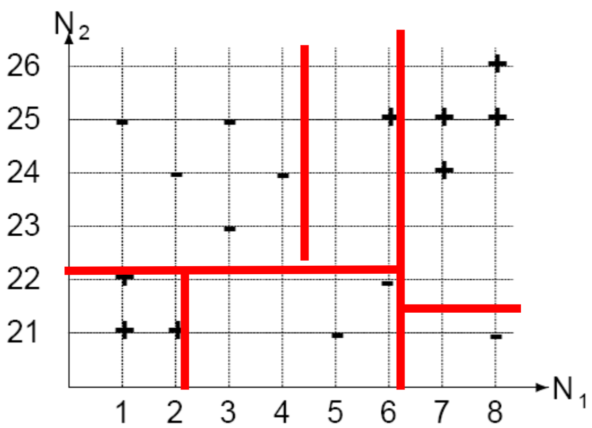
Como vemos, la mejor pregunta sobre N_1 ($N_1>2$) tiene mayor IG que la mejor (única) pregunta sobre N_2, por lo que será la que C4.5 elija. De hecho, $N_1>2$ separa perfectamente las clases: cuando C4.5 expanda recursivamente las dos ramas que cuelgan de esta pregunta (rama $N_1>2$ y rama $N_1\leq 2$), C4.5 detectará que en cada una de ellas hay ejemplos de sólo una clase, por lo que pondrá una hoja en cada una de esas ramas con la clase mayoritaria y terminará su expansión:





Cuando C4.5 analice **D** encontrará que la mejor pregunta es $N_1>4$ (separa completamente las clases), y pone un nodo hoja en cada una de las dos ramas que cuelgan de esta pregunta. Y cuando C4.5 analice **B** encontrará que la mejor pregunta es $N_2>21$ (separa completamente las clases), y pone un nodo hoja en cada una de las dos ramas que cuelgan de esta pregunta.
Con lo cual, el árbol desarrollado por el algoritmo será:





4. Solución:

Patrón 1: 1.2*0 -0.7*0 = 0, que es < 0.3, luego la salida de la neurona es 0
Patrón 2: 1.2*0 -0.7*1 = -0.7, que es < 0.3, luego la salida de la neurona es 0
Patrón 3: 1.2*1 -0.7*0 = 1.2, que es > 0.3, luego la salida de la neurona es 1
Patrón 4: 1.2*1 -0.7*1 = 0.5, que es > 0.3, luego la salida de la neurona es 1

5. Solución:

$h = \sigma(\mathbf{w}^T \cdot \mathbf{x})$
$\Delta\mathbf{w} = -\eta \cdot (h - t) \cdot \mathbf{x}$

$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	t	w_0	w_1	w_2	h	$h-t$	Δw_0	Δw_1	Δw_2
1	0	0	0	-0.3	0	0.4	0.43	0.43	-0.43	0	0
1	1	0	0	-0.73	0	0.4	0.33	0.33	-0.33	-0.33	0
1	0	1	0	-1.06	-0.33	0.4	0.34	0.34	-0.34	0	-0.34
1	1	1	1	-1.4	-0.33	0.06	0.16	-0.84	0.84	0.84	0.84

Se llega tras la primera época a los siguientes pesos:
w0 = -1.4 + 0.84 = -0.56
w1 = -0.33 + 0.84 = 0.51
w2 = 0.06 + 0.84 = 0.90

Extensión a la pregunta original:
¿Cúal sería el resultado de la primera época si en vez de aprendizaje online hubiéramos usado aprendizaje por lotes (batch) usando la misma constante de aprendizaje $\eta = 1$?

$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	t	w_0	w_1	w_2	h	$h-t$	Δw_0	Δw_1	Δw_2
1	0	0	0	-0.3	0	0.4	0.43	0.43	-0.43	0	0
1	1	0	0	-0.3	0	0.4	0.43	0.43	-0.43	-0.43	0
1	0	1	0	-0.3	0	0.4	0.52	0.52	-0.52	0	-0.52
1	1	1	1	-0.3	0	0.4	0.52	-0.48	0.48	0.48	0.48

Luego tras la primera época los pesos quedan del siguiente modo:
w0 = -0.3 - 0.43 - 0.43 - 0.52 + 0.48 = -1.2
w1 = 0 + 0 – 0.43 + 0 + 0.48 = 0.05
w2 = 0.4 + 0 + 0 - 0.52 + 0.48 = 0.36

Nota: el cálculo que hemos realizado en aprendizaje por lotes es exactamente equivalente a la forma en la que viene en las transparencias de clase: calcular primero el gradiente debido a todos los patrones y después actualizar una sola vez los pesos con este gradiente (pasos 2.2 y 2.3 de la transparencia 23 de redes neuronales).