Jesús García Herrero

TÉCNICAS DE INDUCCIÓN-II

En esta clase se continúa con el desarrollo de métodos de inducción de modelos lógicos a partir de datos. Se parte de las limitaciones del método ID3 presentado previamente: tratamiento de atributos con demasiados valores (como fechas), sobreajuste, datos continuos, incompletos, etc. Esto sirve para desarrollar el método C4.5 que extiende el algoritmo ID3 para dar respuesta a esta necesidad, y especialmente buscando mejorar la capacidad de generalización y evitar sobre-ajuste mediante técnicas de poda del árbol resultante. Para ello se utiliza una estimación pesimista del error de clasificación, que aumenta al disminuir el número de datos utilizados en la evaluación, y por tanto favorece modelos más compactos y generales.

A continuación se presenta una segunda estrategia de inducción, mediante cobertura de las clases con las reglas que alcanzan el menor error de clasificación. Se revisan el algoritmo PRISM como representativos del aprendizaje de reglas, con heurísticas de búsqueda con el objetivo indicado. Se presenta a continuación el algoritmo INDUCT que permite mejorar la capacidad de generalización con un proceso de poda que va eliminando las condiciones de las reglas con un heurístico análogo al empleado en C4.5

El tema se completa con una presentación introductoria al problema de la búsqueda de reglas de asociación, como paradigma de aprendizaje no supervisado. Se comenta el algoritmo “A Priori” como representativo de estas técnicas, donde no tenemos un atributo objetivo, como en el problema de clasificación, sino una búsqueda de reglas que identifiquen relaciones con suficiente grado de significatividad.

Método C4.5

Árboles con Algoritmo C4.5

Reglas de Asociación

Jesús García Herrero
Universidad Carlos III de Madrid

Universidad
Carlos III de Madrid

Dificultades con clasificador ID3

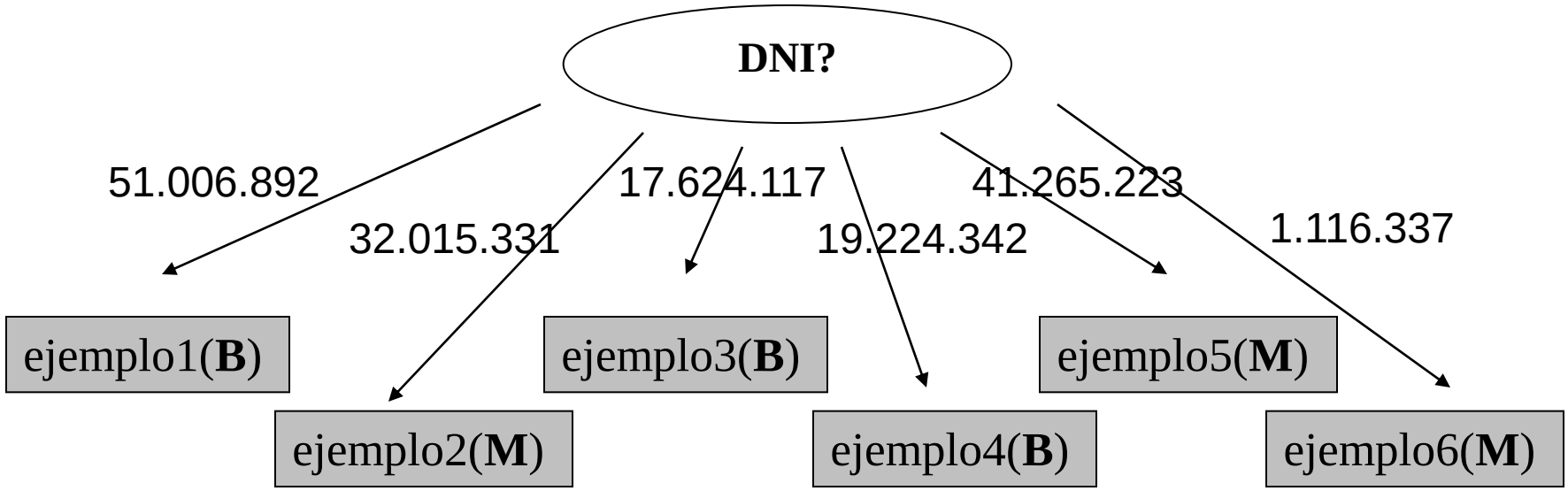
- Qué se hace con valores discretos con muchos valores? d*ía del cumpleaños*
- Cuándo se debe parar de subdividir el árbol? *Sobreadecuación (overfitting), poda*
- Qué se hace con valores continuos de atributos? *discretizar, rangos*
- Qué se hace cuando los ejemplos vienen incrementalmente? *ID4 e ID5*
- Qué ocurre cuando hay atributos con valores desconocidos?
  - asignar el valor más probable
  - asignar distribución de probabilidad

Atributos con muchos valores

- ID3 prefiere atributos con mayor número de valores (problema de código de Identificación)
- Se les puede desfavorecer utilizando la medida Razón de ganancia(GainRatio , GR):

Ejemplo	DNI	Sitio de acceso: A_1	1ª cantidad gastada: A_2	Vivienda: A_3	Última compra: A_4	Clase
1	51.006.892	1	0	2	Libro	Bueno
2	32.015.331	1	0	1	Disco	Malo
3	17.624.117	1	2	0	Libro	Bueno
4	19.224.342	0	2	1	Libro	Bueno
5	41.265.223	1	1	1	Libro	Malo
6	1.116.337	2	2	1	Libro	Malo

Atributos con muchos valores



$$I(DNI) = \sum_{j=1}^{6} \frac{n_{DNIj}}{n} I_{DNIj} = 6\frac{1}{6} I_{DNI} = 0$$

$$I_{DNI} = -\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1} = 0$$

Árboles de clasificación

Atributos con muchos valores

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

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

Atributos con valores continuos

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

PESO	30	36	44	60	72	78
SANO	NO	NO	SI	SI	SI	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 restringe a nodos binarios
  - Se analizan los posibles puntos de corte y se mide la entropía

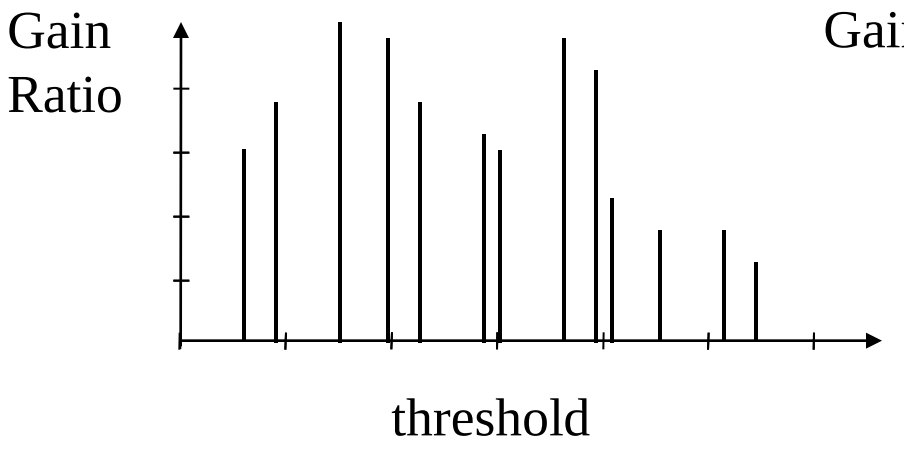
    peso < 40? ; peso < 75?

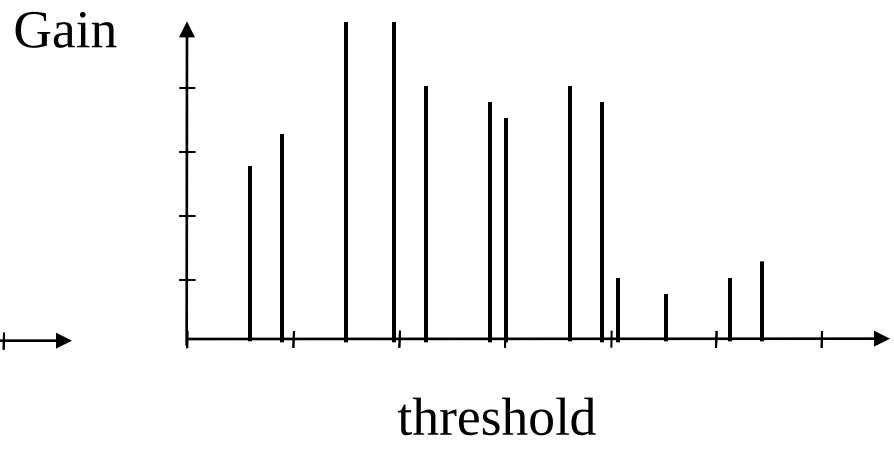
    *I(peso<40)=2/6*0+4/6*0.81=**0.54***

    *I(peso<75)=5/6*0.97+1/6*0=0.81*
  - Otras posibilidades: discretizar el rango de valores, nodos no binarios

Atributos con valores continuos

- Para cada posible atributo continuo: selección del mejor punto de comparación (ganancia de información).
- Puede aparecer varias veces en el árbol, con distintos puntos de corte (umbrales).





Ejemplos con atributos en blanco

- Dos problemas: entrenamiento y clasificación
- Se supone una distribución probabilística del atributo de acuerdo con los valores de los ejemplos en la muestra de entrenamiento
  - Para clasificar: se divide el ejemplo en tantos casos como valores, y se da un peso a cada resultado correspondiente a cada frecuencia
  - Para entrenar: los casos con faltas también se distribuyen con pesos para construir el árbol de decisión.
- El resultado es una clasificación con probabilidades, correspondientes a la distribución de ejemplos en cada nodo hoja

Ejemplo de entrenamiento

OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
sunny	75	70	TRUE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
rainy	75	80	FALSE	yes
rainy	71	91	TRUE	no
overcast	64	65	TRUE	yes
overcast	83	86	FALSE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes

```
outlook = sunny
|   humidity <= 75: yes (2.0)
|   humidity > 75: no (3.0)
outlook = overcast: yes (4.0)
outlook = rainy
|   windy = TRUE: no (2.0)
|   windy = FALSE: yes (3.0)
```

Árboles de clasificación

Entrenamiento con 1 falta

OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
sunny	75	70	TRUE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
rainy	75	80	FALSE	yes
rainy	71	91	TRUE	no
overcast	64	65	TRUE	yes
overcast	83	86	FALSE	yes
?	72	90	TRUE	yes
overcast	81	75	FALSE	yes

Distribución con ejemplos sin faltas

OUTLOOK	PLAY	DON'T PLAY	TOTAL
sunny	2	3	5
overcast	3	0	3
rain	3	2	5
Total	8	5	13

Árboles de clasificación

Entrenamiento instancia con faltas

- Cada ejemplo se pondera con factor w
  - Si el valor del atributo es conocido, w=1
  - Si no, el ejemplo entra en cada rama, con w igual a la probabilidad en ese punto (distribución de ejemplos de entrenamiento)
- En el caso anterior: se “divide” el ejemplo en 3
  - 5/13 a sunny, 3/13 a overcast, 5/13 a rain
- Ej: rama con valor *sunny*

OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
sunny	75	70	TRUE	yes
?	72	90	TRUE	yes

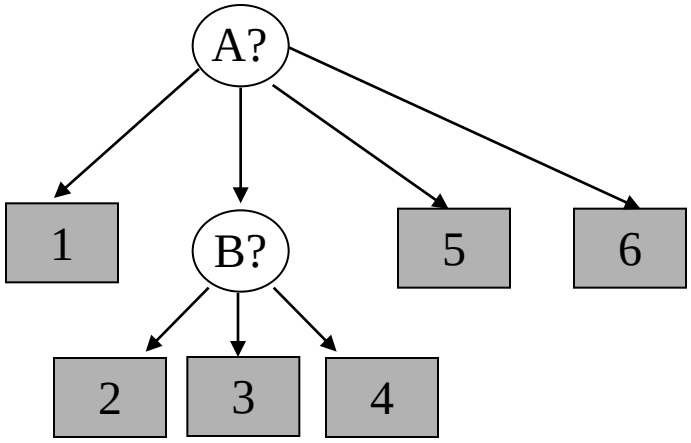
- siguiente atributo con mejor ganancia*: **humidity:***
  - *humidity<=75: 2 ejemplos Play 0 ejemplos Don’t play*
  - *humidity>75: 5/13 ejemplos Play 3 ejemplos Don’t play*

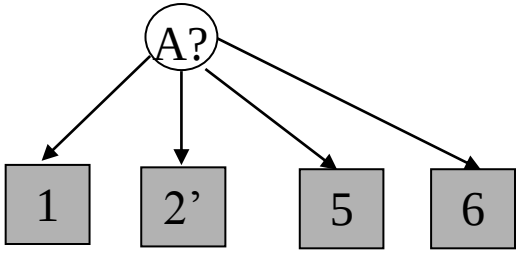
Árbol resultante

	clase yes	clase no
outlook = sunny		
| humidity <= 75: yes (2.0)	100%	0%
| humidity > 75: no (3.38/0.38)	12%	88%
outlook = overcast: yes (3.24)	100%	0%
outlook = rainy		
| windy = TRUE: no (2.38/0.38)	16%	84%
| windy = FALSE: yes (3.0)	100%	0%

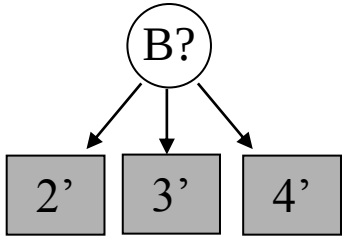
Poda del árbol

- Se hace para evitar el sobre-adecuamiento. Varias posibilidades:
  - **pre-poda**: se decide cuando dejar de subdividir.
  - **post-poda**: se construye el árbol y después se poda. Ventaja de relaciones entre atributos.
- Se consideran dos operaciones de poda:
  - reemplazo de sub-árbol por hoja (*subtree replacement*).
  - elevación de sub-árbol (*subtree raising*).





reemplazo

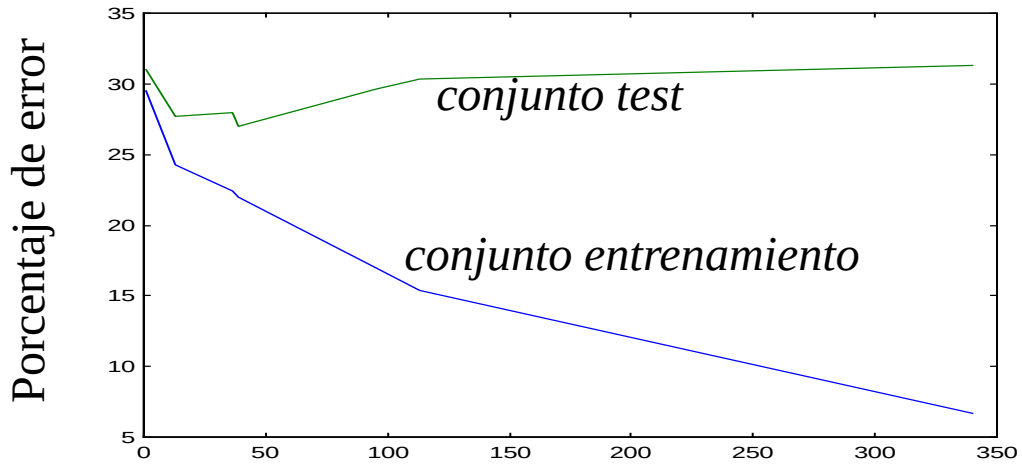


elevación

Árboles de clasificación

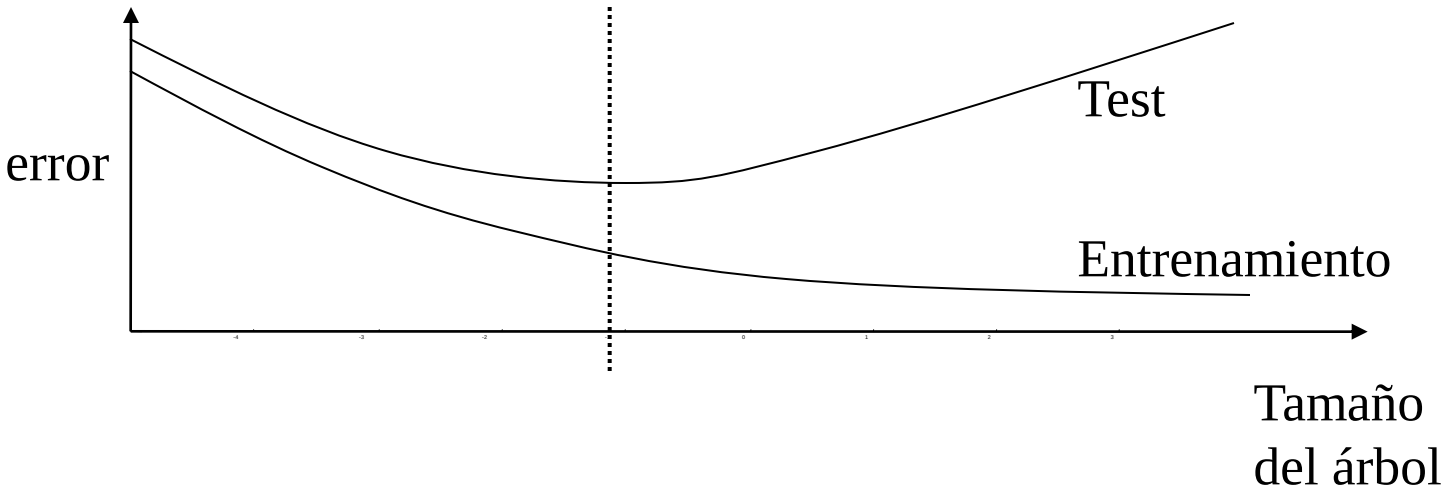
Efecto del tamaño del árbol

Tamaño del árbol	Conjunto de entrenamiento		Conjunto de test	
	instancias incorrectas	porcentaje de error	instancias incorrectas	porcentaje de error
1	207	29.57 %	93	31%
13	170	24.29 %	83	27.67 %
36	157	22.43%	84	28%
39	154	22%	81	27%
95	119	17%	89	29.67%
113	108	15.43%	91	30.3%
340	47	6.71%	94	31.3%



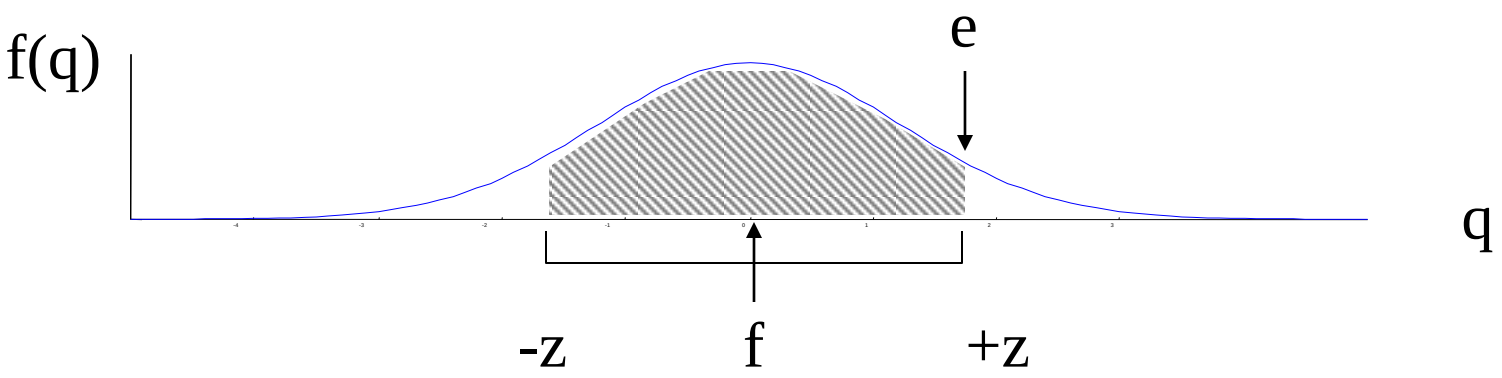
Soluciones pre-poda

- Solución 1: test χ^2 (Quinlan, 86)
  - no se divide un nodo si se tiene poca confianza en él (no es significativa la diferencia de clases). Ej: 40 ejemplos (+) y uno (-)
  - es muy conservador, y puede parar antes de lo conveniente
- Solución 2: Validación con conjunto de test independiente y para cuando la curva del conjunto de test empieza a subir



Soluciones post-poda

- Primera poda: mínimo número de elementos por hoja (sin evaluar)
- Para comparar la mejor solución (podar o no) se debe evaluar cada opción, una vez construido el árbol. Hay dos alternativas
  - Evaluar conjunto de test independiente (reduced error pruning). Desventaja de no aprovechar todos los datos
  - Estimación pesimista de error a partir de error de entrenamiento: f=E/N, q? Extremo superior con un intervalo de confianza α (heurístico)



Estimación de error

$$\mathrm{Prob}\left[\frac{f-q}{\sqrt{q(1-q)/N}} \le z\right] = \alpha$$

$$q = \left(f + \frac{z^2}{2N} + z\sqrt{\frac{f}{N}(1-f) + \frac{z^2}{4N^2}}\right) \Big/ \left(1 + \frac{z^2}{N}\right)$$

- C4.5 utiliza un intervalo de confianza de 25%, que con aproximación gaussiana corresponde a z=0.69
- Algunas versiones no suponen la simplificación normal (distribución binomial)
- Poda de abajo a arriba: cada sub-árbol se evalúa y compara con: 1) una única hoja, 2) rama con más ejemplos

Ejemplo: lentes contacto

Age	spectacle-prescription	astigmatism	tear production rate	contact lenses
young	myope	no	reduced	none
young	myope	no	normal	soft
young	myope	yes	reduced	none
young	myope	yes	normal	hard
young	hypermetrope	no	reduced	none
young	hypermetrope	no	normal	soft
young	hypermetrope	yes	reduced	none
young	hypermetrope	yes	normal	hard
pre-presbyopic	myope	no	reduced	none
pre-presbyopic	myope	no	normal	soft
pre-presbyopic	myope	yes	reduced	none
pre-presbyopic	myope	yes	normal	hard
pre-presbyopic	hypermetrope	no	reduced	none
pre-presbyopic	hypermetrope	no	normal	soft
pre-presbyopic	hypermetrope	yes	reduced	none
pre-presbyopic	hypermetrope	yes	normal	none
presbyopic	myope	no	reduced	none
presbyopic	myope	no	normal	none
presbyopic	myope	yes	reduced	none
presbyopic	myope	yes	normal	hard
presbyopic	hypermetrope	no	reduced	none
presbyopic	hypermetrope	no	normal	soft
presbyopic	hypermetrope	yes	reduced	none
presbyopic	hypermetrope	yes	normal	none

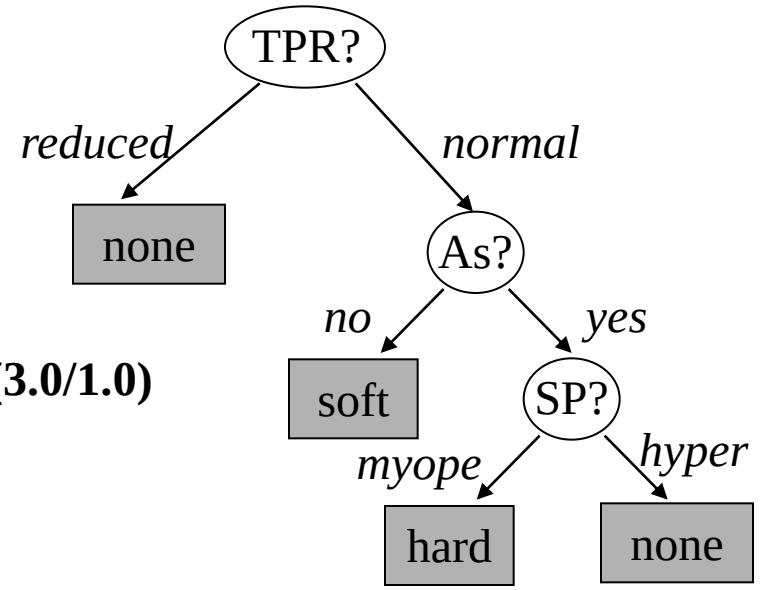
Árboles de clasificación

Árbol original

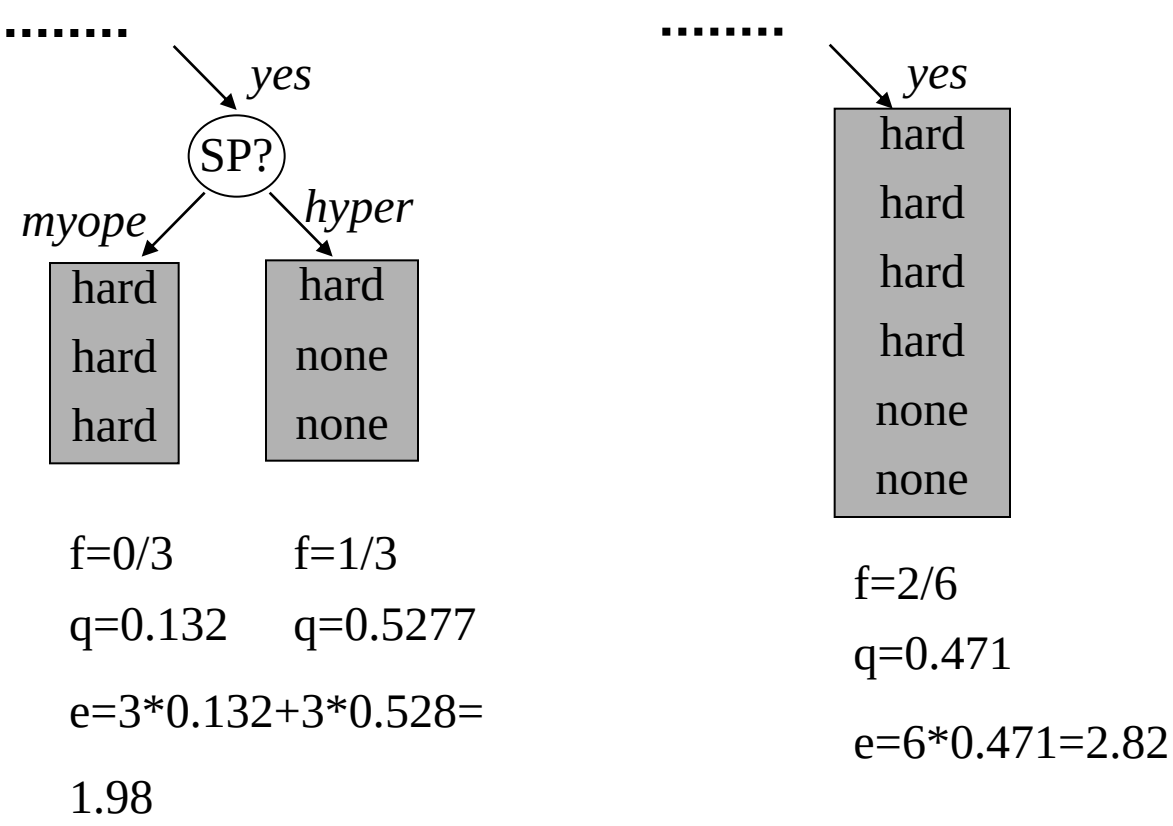
```
tear-prod-rate = reduced: none (12.0)
tear-prod-rate = normal
|   astigmatism = no
|   |   age = young: soft (2.0)
|   |   age = pre-presbyopic: soft (2.0)
|   |   age = presbyopic
|   |   |   spectacle-prescrip = myope: none (1.0)
|   |   |   spectacle-prescrip = hypermetrope: soft (1.0)
|   astigmatism = yes
|   |   spectacle-prescrip = myope: hard (3.0)
|   |   spectacle-prescrip = hypermetrope
|   |   |   age = young: hard (1.0)
|   |   |   age = pre-presbyopic: none (1.0)
|   |   |   age = presbyopic: none (1.0)
```

Poda1: 2 Ejemplos por hoja

tear-prod-rate = reduced: none (12.0)
tear-prod-rate = normal
| astigmatism = no: soft (6.0/1.0)
| astigmatism = yes
| | spectacle-prescrip = myope: hard (3.0)
| | spectacle-prescrip = hypermetrope: none (3.0/1.0)



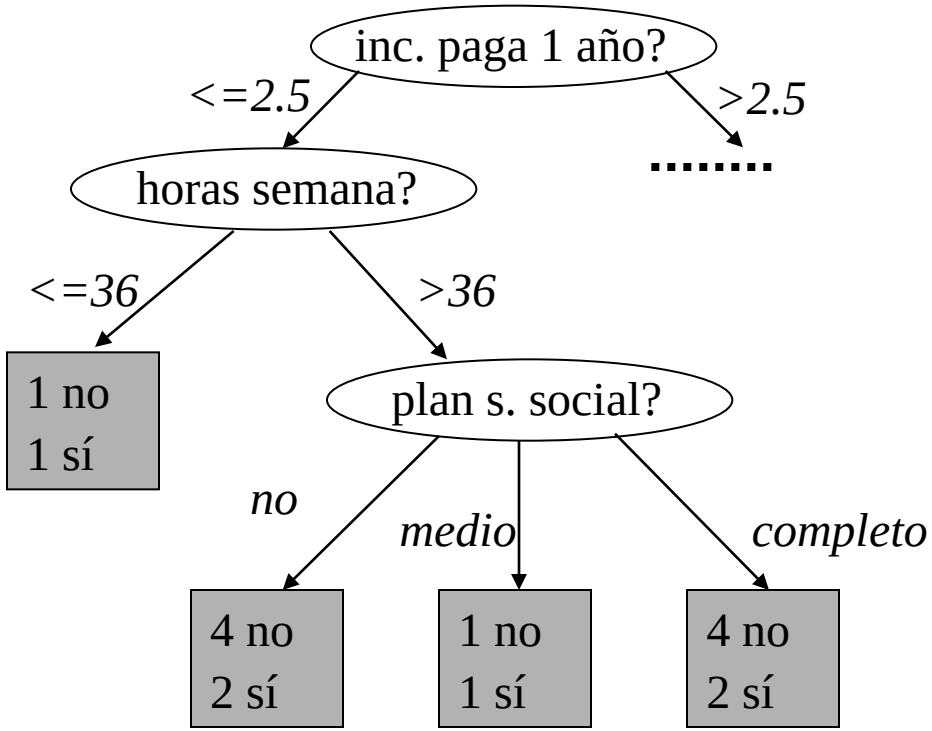
Poda2: Mejora el error?



No mejora el error al reemplazar sub-árbol por hoja

Árboles de clasificación

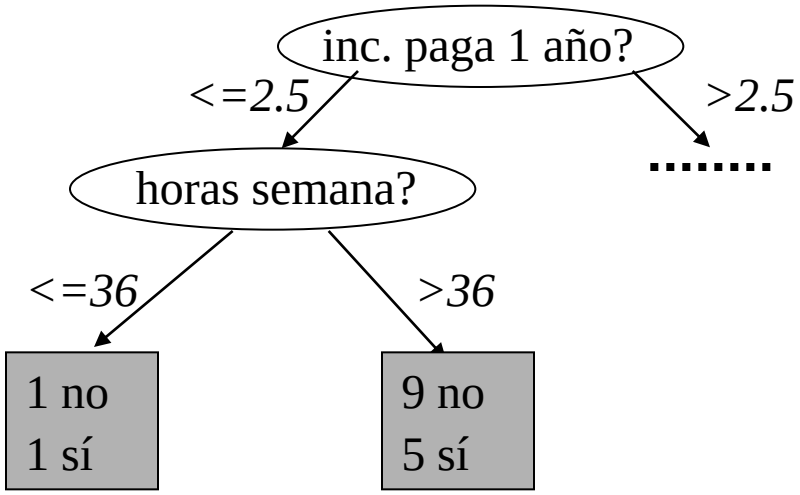
Ejemplo 2: negociación laboral



f=2/6 f=1/2 f=2/6

q=0.47 q=0.72 q=0.47

e=6*0.47+2*0.72+6*0.47=7.08



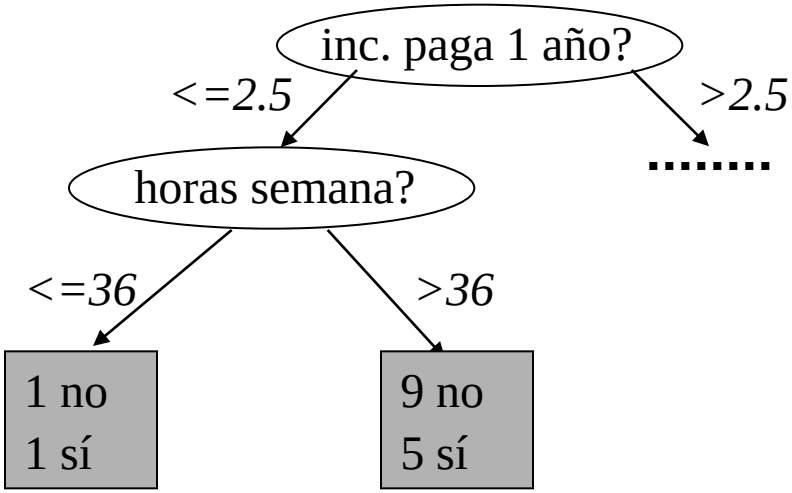
f=5/14

q=0.447

e=14*0.447=6.25

Árboles de clasificación

Ejemplo 2: segunda poda



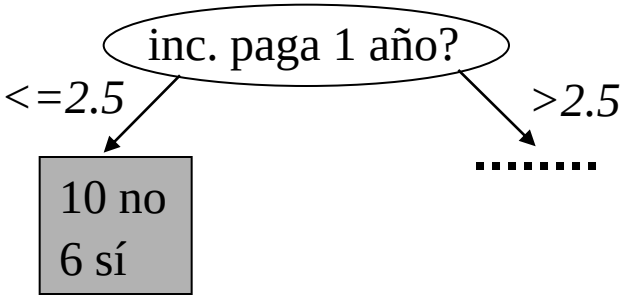
f=1/2

q=0.72

f=5/14

q=0.446

e=2*0.72+14*0.446=7.69



f=6/16

q=0.459

e=16*0.459=7.34

Estimación pesimista de error

horas semana <= 36: SÍ (2.0/1.44)
horas semana >36
| plan s. social = no: NO (6.0/2.82)
| plan s. social = medio: SÍ (2.0/1.44)
| plan s. social = completo: NO (6.0/2.82)|

subárbol podado (1ª poda)
horas semana <= 36: SÍ (2.0/1.44)
horas semana >36: NO(14.0/6.25)

subárbol podado (2ª poda)
NO (16.0/7.34)

Dificultades con clasificador ID3

- Qué se hace con valores discretos con muchos valores? d*ía del cumpleaños*
- Cuándo se debe parar de subdividir el árbol? *Sobreadecuación (overfitting), poda*
- Qué se hace con valores continuos de atributos? *discretizar, rangos*
- Qué se hace cuando los ejemplos vienen incrementalmente? *ID4 e ID5*
- Qué ocurre cuando hay atributos con valores desconocidos?
  - asignar el valor más probable
  - asignar distribución de probabilidad

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
SANO	NO	NO	SI	SI	SI	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 restringe a nodos binarios
  - Se analizan los posibles puntos de corte y se mide la entropía

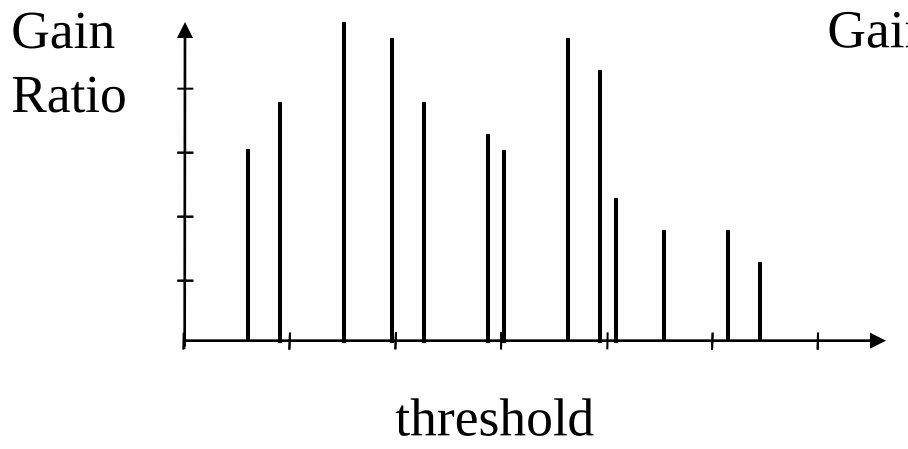
    peso < 40? ; peso < 75?

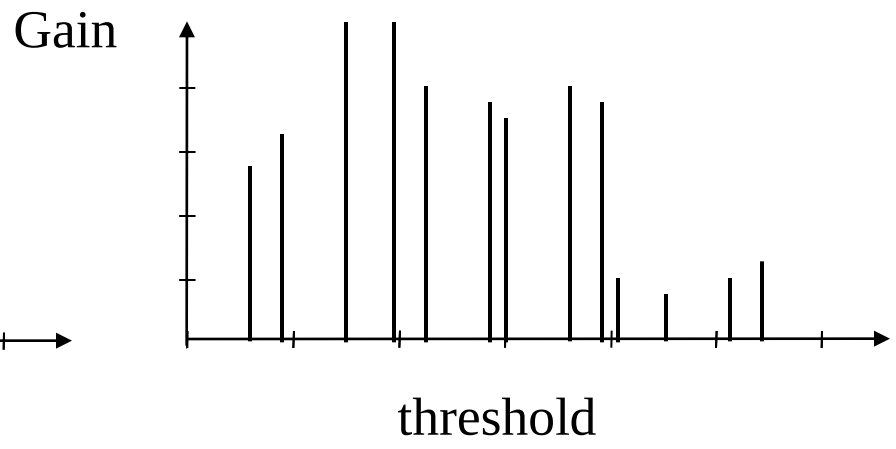
    *I(peso<40)=2/6*0+4/6*0.81=**0.54***

    *I(peso<75)=5/6*0.97+1/6*0=0.81*
  - Otras posibilidades: discretizar el rango de valores, nodos no binarios

Atributos con valores continuos

- Para cada posible atributo continuo: selección del mejor punto de comparación (ganancia de información).
- Puede aparecer varias veces en el árbol, con distintos puntos de corte (umbrales).





Ejemplos con atributos en blanco

- Dos problemas: entrenamiento y clasificación
- Se supone una distribución probabilística del atributo de acuerdo con los valores de los ejemplos en la muestra de entrenamiento
  - Para clasificar: se divide el ejemplo en tantos casos como valores, y se da un peso a cada resultado correspondiente a cada frecuencia
  - Para entrenar: los casos con faltas también se distribuyen con pesos para construir el árbol de decisión.
- El resultado es una clasificación con probabilidades, correspondientes a la distribución de ejemplos en cada nodo hoja

Ejemplo de entrenamiento

OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
sunny	75	70	TRUE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
rainy	75	80	FALSE	yes
rainy	71	91	TRUE	no
overcast	64	65	TRUE	yes
overcast	83	86	FALSE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes

outlook = sunny
| humidity <= 75: yes (2.0)
| humidity > 75: no (3.0)
outlook = overcast: yes (4.0)
outlook = rainy
| windy = TRUE: no (2.0)
| windy = FALSE: yes (3.0)

Clasificación con C4.5

Entrenamiento con 1 falta

OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
sunny	75	70	TRUE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
rainy	75	80	FALSE	yes
rainy	71	91	TRUE	no
overcast	64	65	TRUE	yes
overcast	83	86	FALSE	yes
?	72	90	TRUE	yes
overcast	81	75	FALSE	yes

Distribución con ejemplos sin faltas

OUTLOOK	PLAY	DON'T PLAY	TOTAL
sunny	2	3	5
overcast	3	0	3
rain	3	2	5
Total	8	5	13

Clasificación con C4.5

Entrenamiento instancia con faltas

- Cada ejemplo se pondera con factor w
  - Si el valor del atributo es conocido, w=1
  - Si no, el ejemplo entra en cada rama, con w igual a la probabilidad en ese punto (distribución de ejemplos de entrenamiento)
- En el caso anterior: se “divide” el ejemplo en 3
  - 5/13 a sunny, 3/13 a overcast, 5/13 a rain
- Ej: rama con valor *sunny*

OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
sunny	75	70	TRUE	yes
?	72	90	TRUE	yes

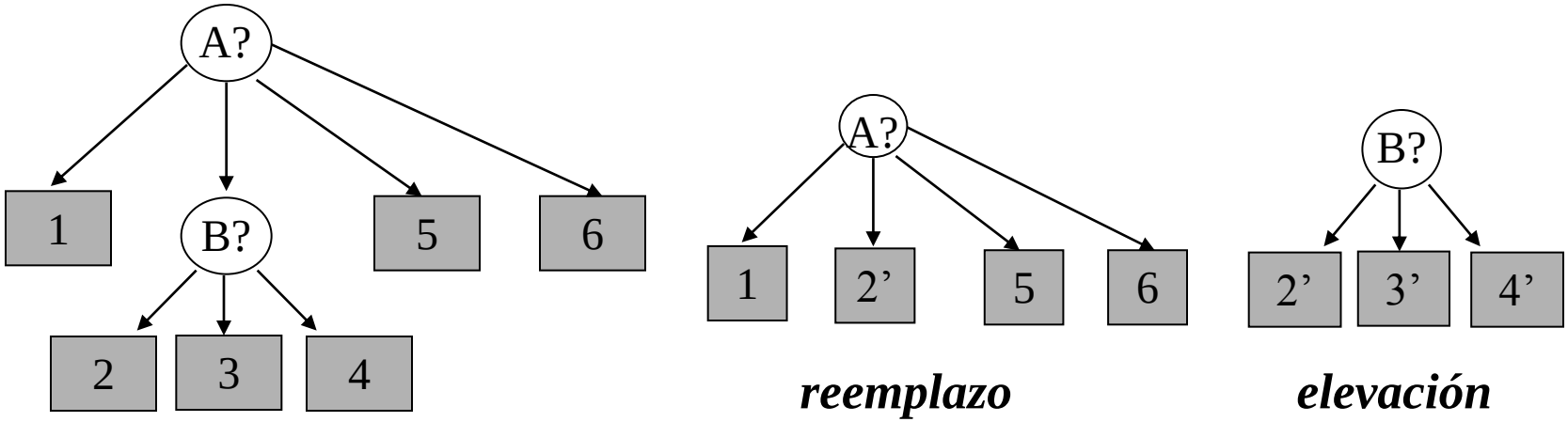
- siguiente atributo con mejor ganancia*: **humidity:***
  - *humidity<=75: 2 ejemplos Play 0 ejemplos Don’t play*
  - *humidity>75: 5/13 ejemplos Play 3 ejemplos Don’t play*

Árbol resultante

	clase yes	clase no
outlook = sunny		
| humidity <= 75: yes (2.0)	100%	0%
| humidity > 75: no (3.38/0.38)	12%	88%
outlook = overcast: yes (3.24)	100%	0%
outlook = rainy		
| windy = TRUE: no (2.38/0.38)	16%	84%
| windy = FALSE: yes (3.0)	100%	0%

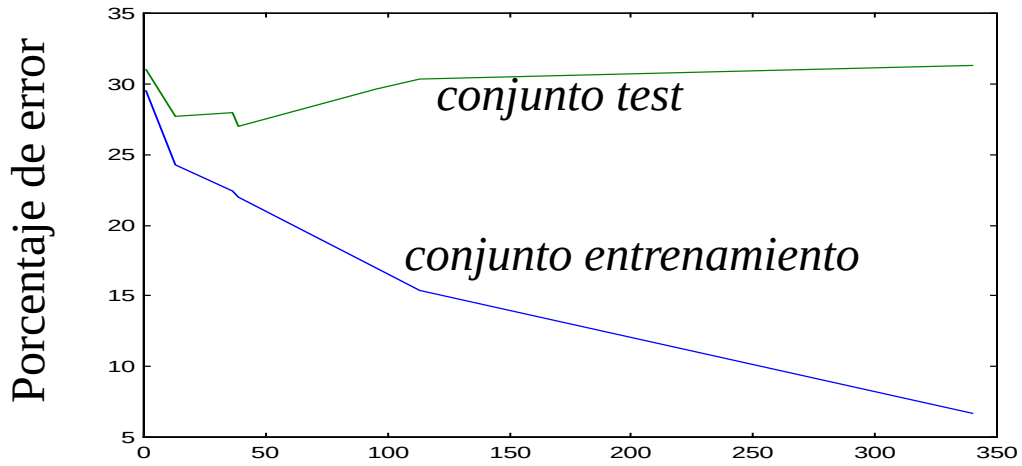
Poda del árbol

- Se hace para evitar el sobre-adecuamiento. Varias posibilidades:
  - **pre-poda**: se decide cuando dejar de subdividir.
  - **post-poda**: se construye el árbol y después se poda. Ventaja de relaciones entre atributos.
- Se consideran dos operaciones de poda:
  - reemplazo de sub-árbol por hoja (*subtree replacement*).
  - elevación de sub-árbol (*subtree raising*).



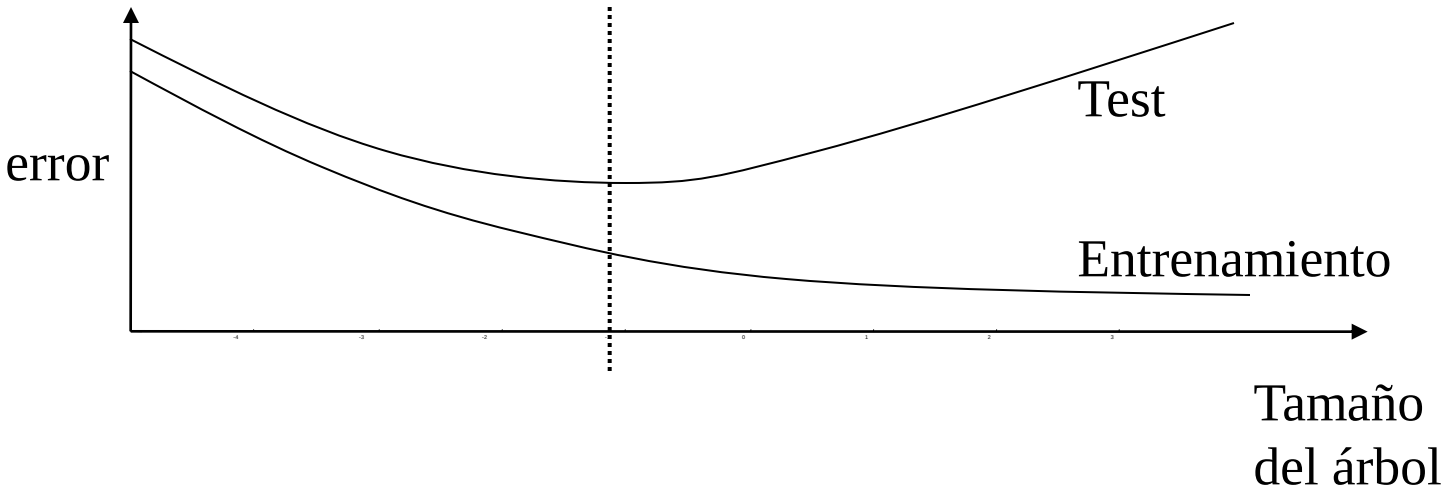
Efecto del tamaño del árbol

Tamaño del árbol	Conjunto de entrenamiento		Conjunto de test	
	instancias incorrectas	porcentaje de error	instancias incorrectas	porcentaje de error
1	207	29.57 %	93	31%
13	170	24.29 %	83	27.67 %
36	157	22.43%	84	28%
39	154	22%	81	27%
95	119	17%	89	29.67%
113	108	15.43%	91	30.3%
340	47	6.71%	94	31.3%



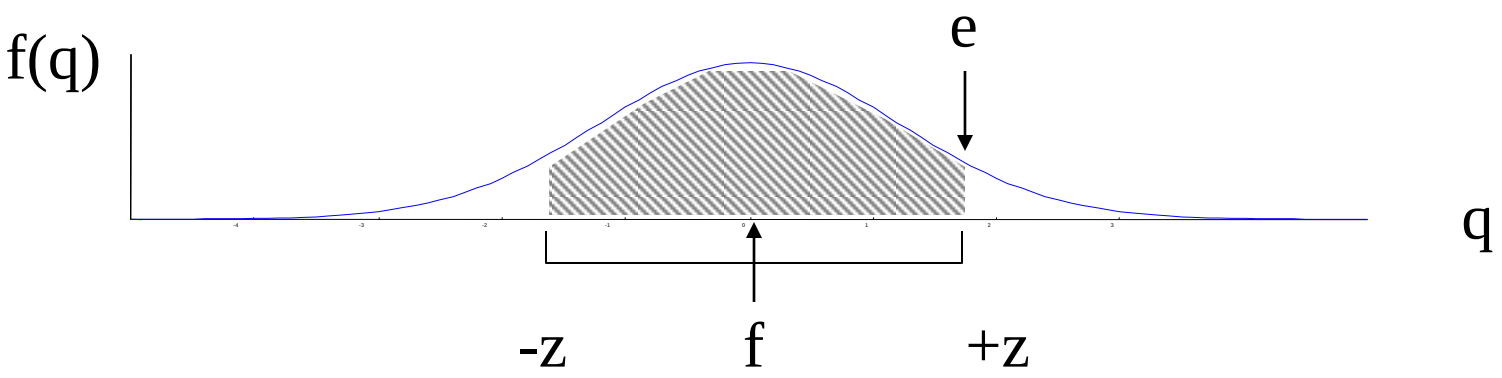
Soluciones pre-poda

- Solución 1: test χ^2 (Quinlan, 86)
  - no se divide un nodo si se tiene poca confianza en él (no es significativa la diferencia de clases). Ej: 40 ejemplos (+) y uno (-)
  - es muy conservador, y puede parar antes de lo conveniente
- Solución 2: Validación con conjunto de test independiente y para cuando la curva del conjunto de test empieza a subir



Soluciones post-poda

- Primera poda: mínimo número de elementos por hoja (sin evaluar)
- Para comparar la mejor solución (podar o no) se debe evaluar cada opción, una vez construido el árbol. Hay dos alternativas
  - Evaluar conjunto de test independiente (reduced error pruning). Desventaja de no aprovechar todos los datos
  - Estimación pesimista de error a partir de error de entrenamiento: f=E/N, q? Extremo superior con un intervalo de confianza α (heurístico)



Estimación de error

$$\mathrm{Prob}\left[\frac{f-q}{\sqrt{q(1-q)/N}} \le z\right] = \alpha$$

$$q = \left(f + \frac{z^2}{2N} + z\sqrt{\frac{f}{N}(1-f) + \frac{z^2}{4N^2}}\right) \Big/ \left(1 + \frac{z^2}{N}\right)$$

- C4.5 utiliza un intervalo de confianza de 25%, que con aproximación gaussiana corresponde a z=0.69
- Algunas versiones no suponen la simplificación normal (distribución binomial)
- Poda de abajo a arriba: cada sub-árbol se evalúa y compara con: 1) una única hoja, 2) rama con más ejemplos

Clasificación con C4.5

Ejemplo: lentes contacto

Age	spectacle-prescription	astigmatism	tear production rate	contact lenses
young	myope	no	reduced	none
young	myope	no	normal	soft
young	myope	yes	reduced	none
young	myope	yes	normal	hard
young	hypermetrope	no	reduced	none
young	hypermetrope	no	normal	soft
young	hypermetrope	yes	reduced	none
young	hypermetrope	yes	normal	hard
pre-presbyopic	myope	no	reduced	none
pre-presbyopic	myope	no	normal	soft
pre-presbyopic	myope	yes	reduced	none
pre-presbyopic	myope	yes	normal	hard
pre-presbyopic	hypermetrope	no	reduced	none
pre-presbyopic	hypermetrope	no	normal	soft
pre-presbyopic	hypermetrope	yes	reduced	none
pre-presbyopic	hypermetrope	yes	normal	none
presbyopic	myope	no	reduced	none
presbyopic	myope	no	normal	none
presbyopic	myope	yes	reduced	none
presbyopic	myope	yes	normal	hard
presbyopic	hypermetrope	no	reduced	none
presbyopic	hypermetrope	no	normal	soft
presbyopic	hypermetrope	yes	reduced	none
presbyopic	hypermetrope	yes	normal	none

Clasificación con C4.5

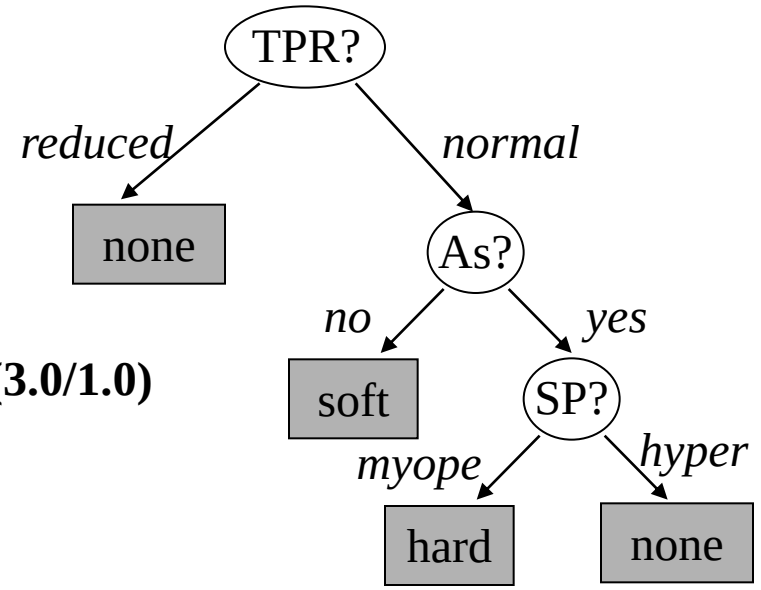
Árbol original

tear-prod-rate = reduced: none (12.0)
tear-prod-rate = normal
| astigmatism = no
| | age = young: soft (2.0)
| | age = pre-presbyopic: soft (2.0)
| | age = presbyopic
| | | spectacle-prescrip = myope: none (1.0)
| | | spectacle-prescrip = hypermetrope: soft (1.0)
| astigmatism = yes
| | spectacle-prescrip = myope: hard (3.0)
| | spectacle-prescrip = hypermetrope
| | | age = young: hard (1.0)
| | | age = pre-presbyopic: none (1.0)
| | | age = presbyopic: none (1.0)

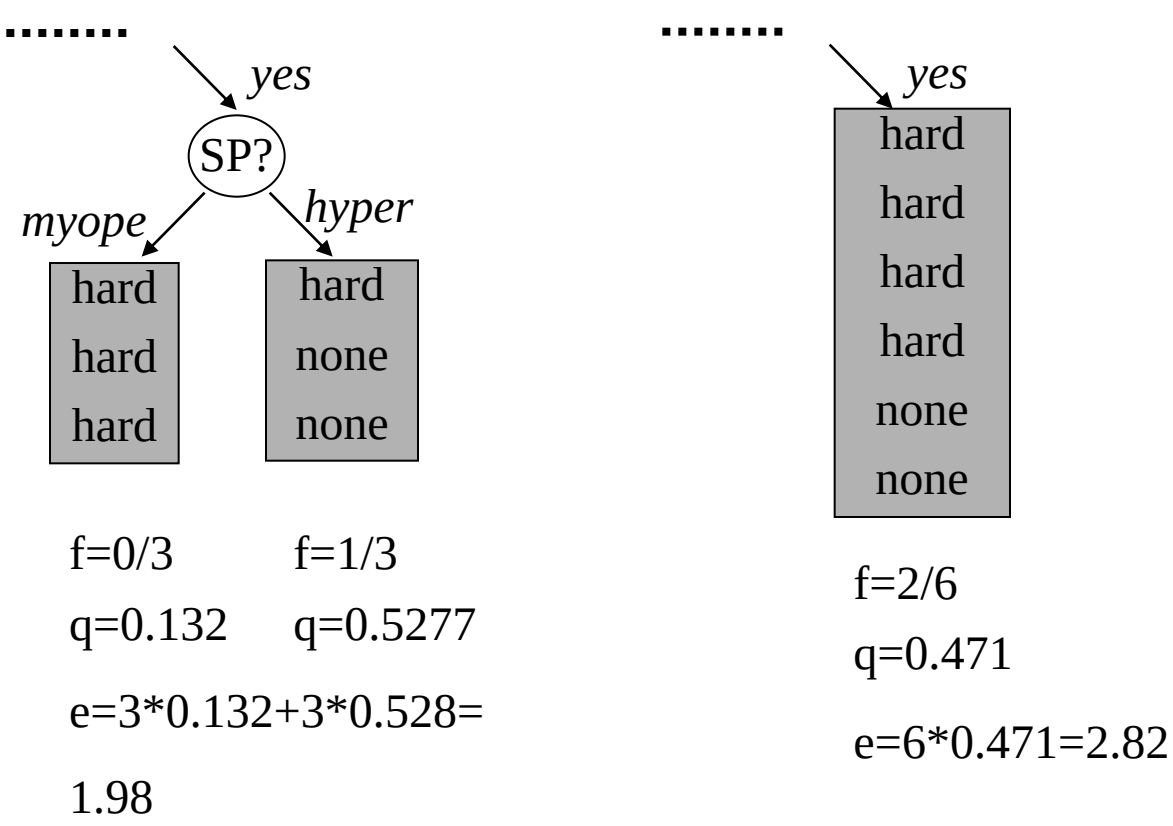
Clasificación con C4.5

Poda1: 2 Ejemplos por hoja

tear-prod-rate = reduced: none (12.0)
tear-prod-rate = normal
| astigmatism = no: soft (6.0/1.0)
| astigmatism = yes
| | spectacle-prescrip = myope: hard (3.0)
| | spectacle-prescrip = hypermetrope: none (3.0/1.0)



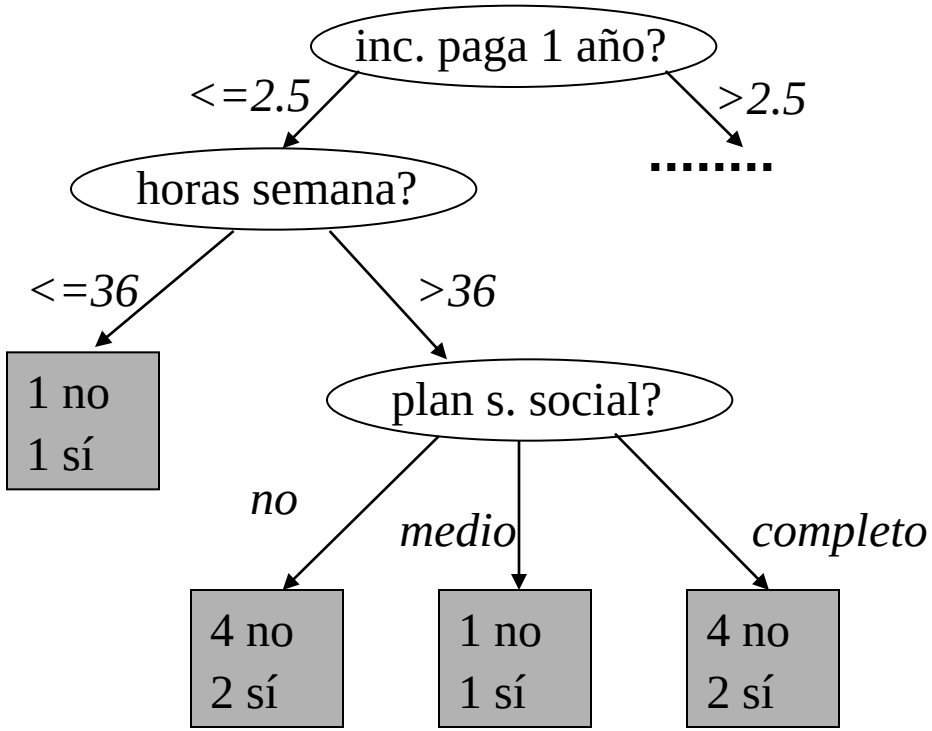
Poda2: Mejora el error?



No mejora el error al reemplazar sub-árbol por hoja

Clasificación con C4.5

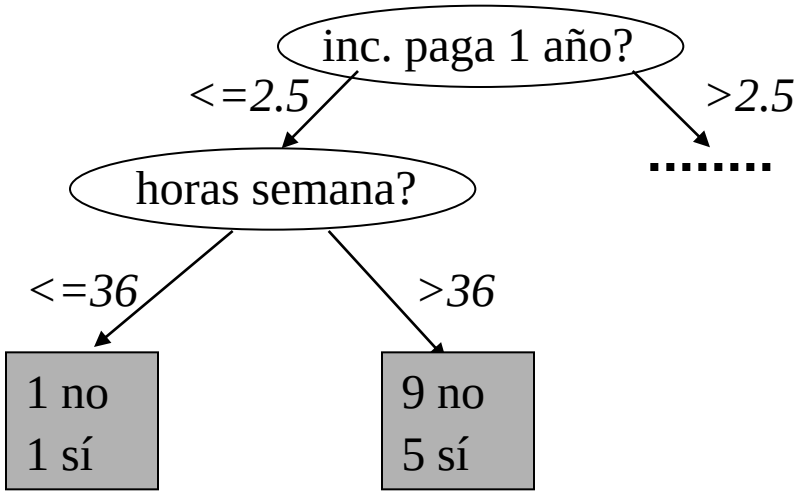
Ejemplo 2: negociación laboral



f=2/6 f=1/2 f=2/6

q=0.47 q=0.72 q=0.47

e=6*0.47+2*0.72+6*0.47=7.08



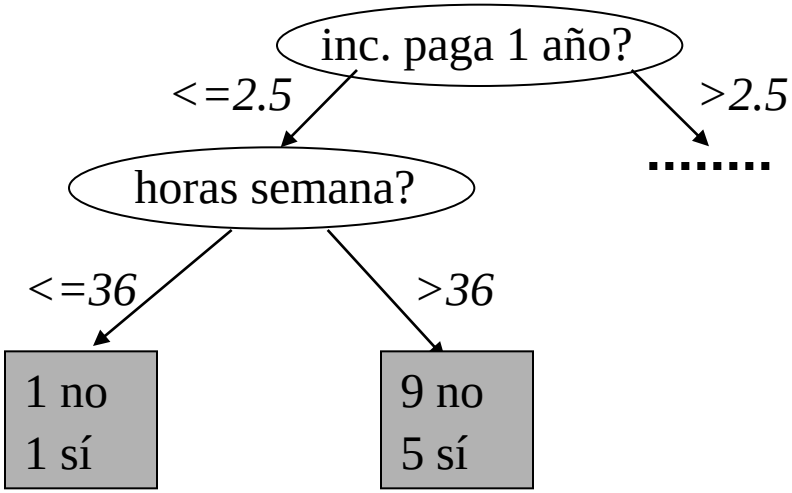
f=5/14

q=0.447

e=14*0.447=6.25

Clasificación con C4.5

Ejemplo 2: segunda poda



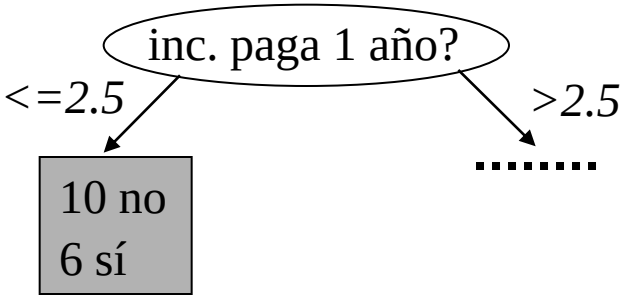
f=1/2

q=0.72

f=5/14

q=0.446

e=2*0.72+14*0.446=7.69



f=6/16

q=0.459

e=16*0.459=7.34

Estimación pesimista de error

horas semana <= 36: SÍ (2.0/1.44)
horas semana >36
| plan s. social = no: NO (6.0/2.82)
| plan s. social = medio: SÍ (2.0/1.44)
| plan s. social = completo: NO (6.0/2.82)|

subárbol podado (1ª poda)
horas semana <= 36: SÍ (2.0/1.44)
horas semana >36: NO(14.0/6.25)

subárbol podado (2ª poda)
NO (16.0/7.34)

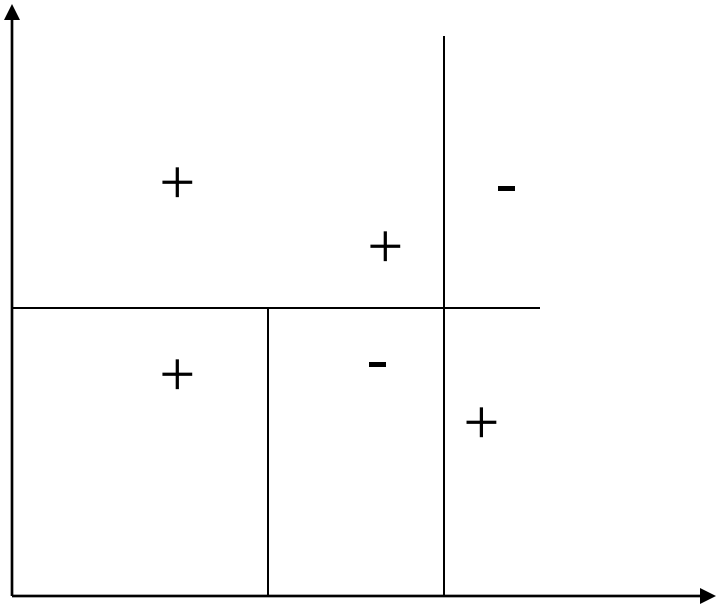
Clasificación con C4.5

Aprendizaje de reglas

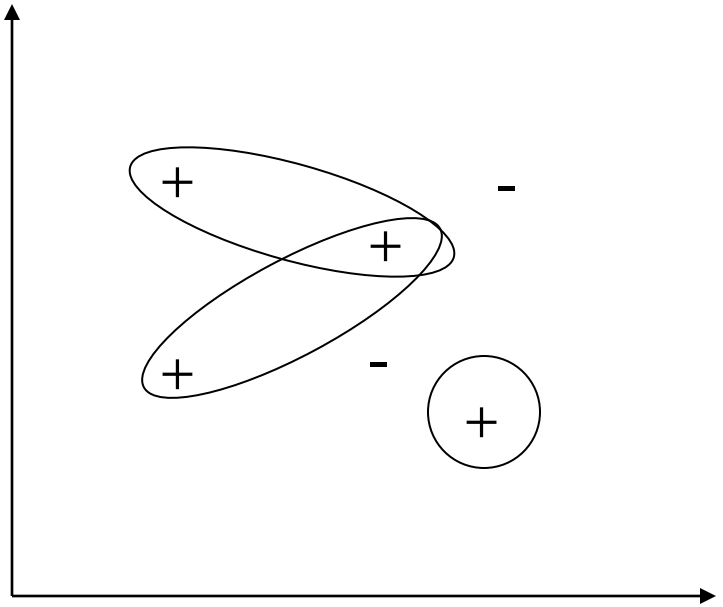
- Objetivos.
  - Dados:
    - Un conjunto de ejemplos de distingas clases (ej.: + y -)
    - Una lista de criterios de preferencia de reglas
  - Determinar: un conjunto de reglas generales que engloban a todos los ejemplos positivos y a ninguno o muy pocos negativos.

- Metodología de cobertura: “covering” o “separate and conquer”
  - Con dos clases: puede usarse hipótesis de “universo cerrado”
  - Con varias clases: reglas separadas para cada clase: reglas independientes: AQ, PRISM
  - Para evitar sobreajuste: poda de reglas: INDUCT
    - criterios heurísticos de evaluación de calidad de una regla
    - se pierde la independencia entre reglas: listas de decisión

Divide y vencerás vs Separa y vencerás



¿Qué atributo(s) separa(n) mejor todos los ejemplos?



¿Qué regla describe mejor cada subconjunto?

Aprendizaje de reglas

- Criterios de preferencias: regla A=>B
  - **Cobertura** (*support*): numero de ejemplos descritos por la regla de todo el espacio de instancias dividido por el numero de ejemplos total. s=|ejemplos con A y B| / |ejemplos total|
  - **Precisión**: ejemplos positivos dividido por ejemplos cubiertos en antecedente: s=|ejemplos con A y B| / |ejemplos con A|
  - **Simplicidad**: numero de preguntas por atributos que aparecen en las condiciones de la regla.
  - **Coste**: suma de los costes de calcular el valor de los atributos de la regla.

Algoritmo PRISM

- Estrategia de cobertura con búsqueda subóptima (alg. voraz)

Para cada clase C

- *Inicializar E al conjunto de instancias*
- *Mientras E contenga instancias de clase C*
  - *Crear regla con antecedente vacío que prediga clase C*
  - *Hasta que R es perfecta (o no hay más atributos)*
    - *Para cada atributo A, no incluido en R, y cada valor V,*
      - *Considerar añadir la condición A=v a la regla*
      - *Seleccionar A y V que maximicen la precisión (p/t). (un empate se resuelve con la condición máxima cobertura t).*
    - *Añadir A=V a R*
  - *Borrar las instancias cubiertas por R de E*
- Reglas independientes del orden de evaluación

Aprendizaje de reglas

Algoritmo AQ (+ y -)

Algoritmo de búsqueda en árbol (optimización global con retroceso):

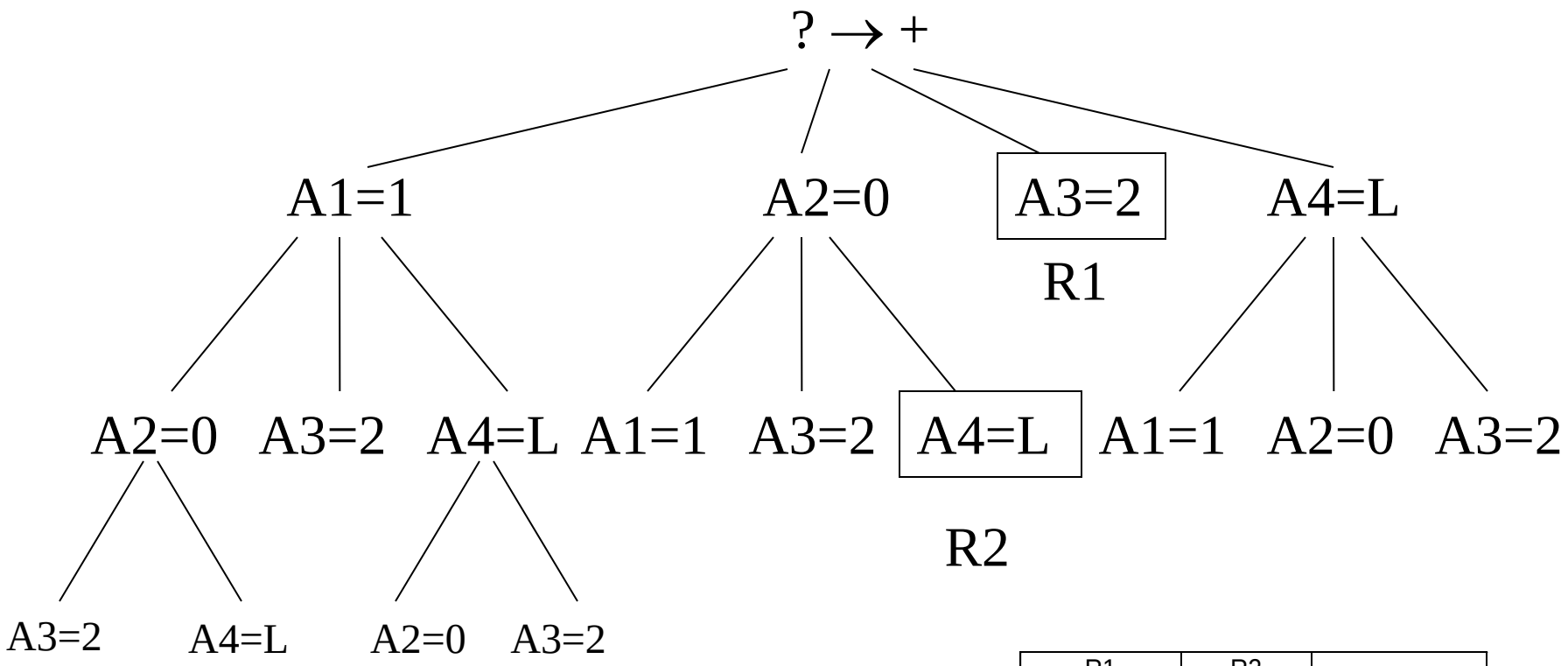
1. *Se elige un ejemplo positivo (semilla)*
2. *Se genera un conjunto de reglas generales (estrella) que describan el ejemplo positivo y no describan ningún ejemplo negativo. Esto se realiza de arriba hacia abajo desde la regla más general (búsqueda en árbol)*
3. *Se selecciona una regla de acuerdo a los criterios de preferencia*
4. *Si la regla seleccionada junto a todas las reglas seleccionadas anteriormente cubre todos los ejemplos positivos, se para*
5. *Si no, separar ejemplos ya cubiertos y volver a 1*

Ejemplo

Sitio de acceso: A_1	1ª cantidad gastada: A_2	Vivienda: A_3	Última compra: A_4	Clase
1	0	2	Libro	+
1	0	1	Disco	-
1	2	0	Libro	+
0	2	1	Libro	+
1	1	1	Libro	-
2	2	1	Libro	-

AQ

Primer ejemplo positivo: 1 0 2 L +

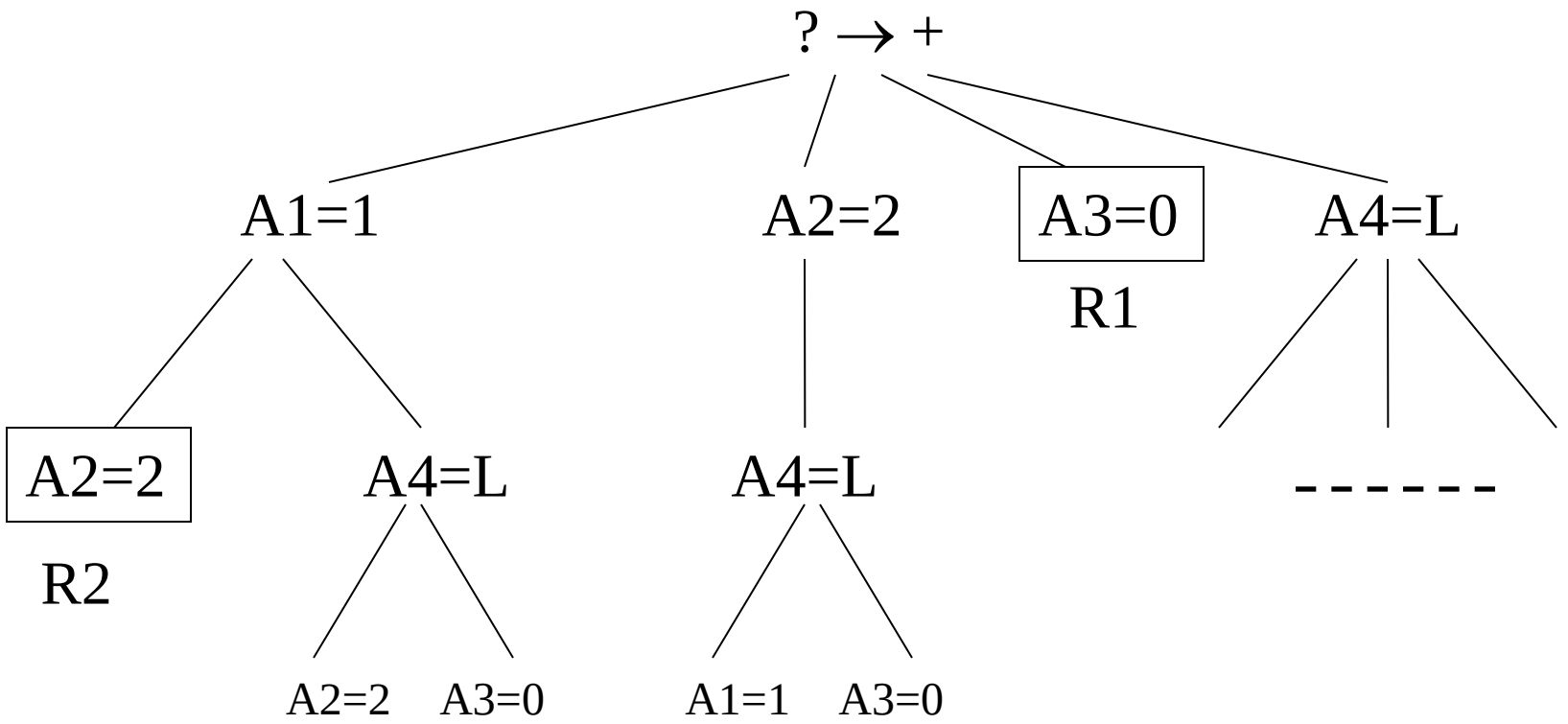


	R1	R2	
Cobertura	1	1	
Simplicidad	1	3	

Estrella: { A3=2 → + }

AQ

Siguiente ejemplo: 1 2 0 L +

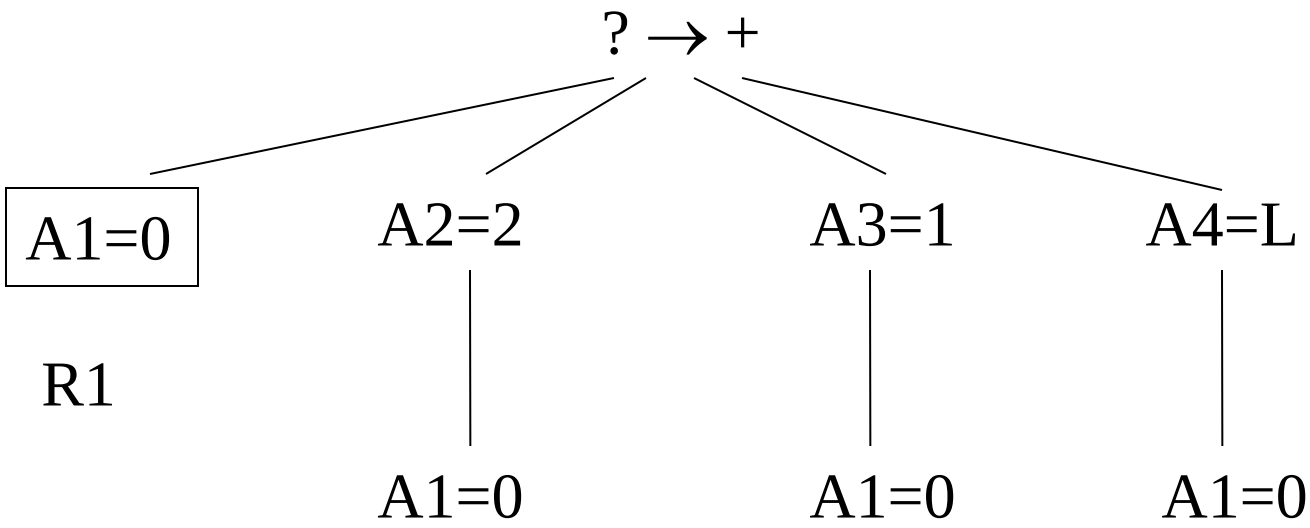


Estrella: { A3=2 → + , A3=0 → + }

	R1	R2
Cobertura	1	1
Simplicidad	1	2

AQ

Siguiente ejemplo: 0 2 1 L +



Estrella: { A3=2 → + , A3=0 → +, A1=0 → +}

Aprendizaje de reglas

PRISM

- Clase: +*:* **si (?) entonces +** (3 ejemplos positivos en 6)
- Opciones:

A1=0	2/4
A1=1	1/1
A2=0	1/2
A2=2	2/3
A3=0	1/1
A3=1	1/4
A3=2	1/1
A4=Libro	3/5

Regla 1: A3=2->+

PRISM

- Clase: +: **si (?) entonces +** (2 ejemplos positivos en 5)
- Opciones:

A1=0	1/1
A1=1	1/3
A2=2	2/3
A3=0	1/1
A3=1	1/4
A4=Libro	2/4

Regla 2: A3=0->+

PRISM

- Clase: +: **si (?) entonces +** (1 ejemplos positivos en 4)
- opciones

A1=1	1/1
A2=2	1/2
A3=1	1/3
A4=Libro	1/3

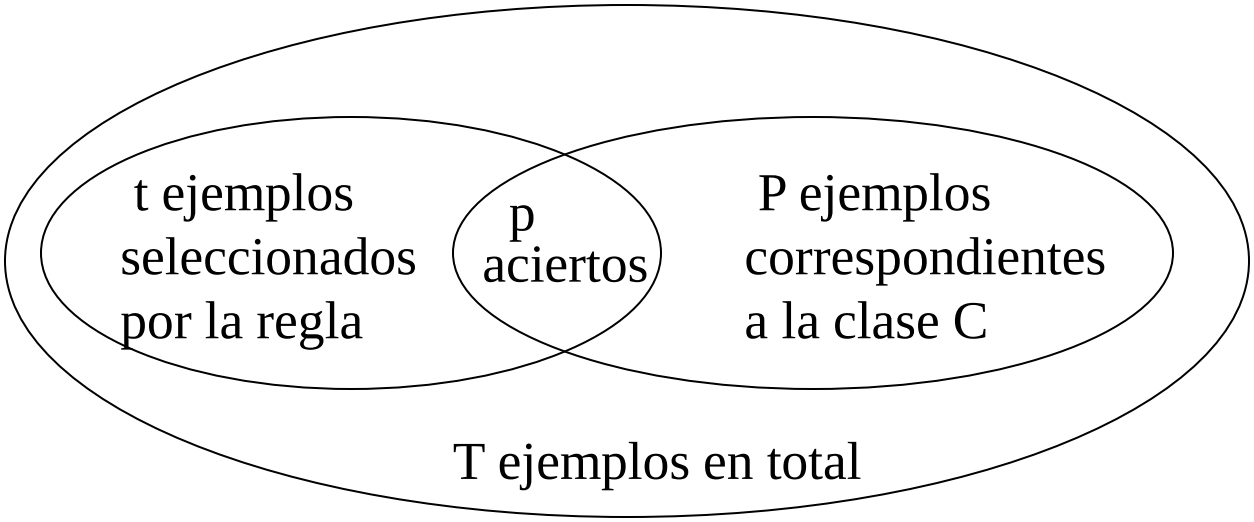
Regla 3: A1=1->+

Evaluación de reglas

- Las reglas “perfectas” asumen falta de ruido y pueden llevar a sobreajuste (*overfitting*)
- Conviene relajar la restricción de ejemplos de otras clases (errores) cubiertos por la regla en construcción
- Dada una regla que cubre t ejemplos de los T existentes, con p predichos correctamente de entre los P correspondientes a la clase:
  - Precisión de la regla: p/t
  - Proporción en el conjunto: P/T
  - Una regla con un solo caso cubierto, maximiza p/t, ¿cobertura p/T es suficiente?

- Evaluación probabilística del valor de una regla
  - Probabilidad de que, tomando t ejemplos al azar, la cantidad de aciertos sea igual o superior a los de la regla: p

Evaluación de reglas



- P(i,t)=Prob(seleccionar t ejemplos que contengan i de clase C)
  - De los t ejemplos, i pertenecen a P, y t-i a T-P:
    - Combinaciones posibles: $\binom{P}{i}\binom{T-P}{t-i}$
  - Posibles grupos de t ejemplos extraídos de T: $\binom{T}{t}$

Evaluación de reglas

- Valor de la regla: probabilidad de mejorar el resultado seleccionando t elementos al azar:
  - Val(R)=P(p,t)+P(p+1,t)+...+P(u,t)
  - u=minimo(t,P)

$$\mathrm{Val}(R) = \frac{1}{\binom{T}{t}} \sum_{i=p}^{\min(P,t)} \binom{P}{i}\binom{T-P}{t-i}$$

Algoritmo con poda (INDUCT)

- *Inicializar E al conjunto de ejemplos*
- *Mientras E no esté vacío*
  - *Para cada clase C representada en E*
    - *Generar la mejor regla "perfecta" para esa clase, R*
    - *Calcular la medida probabilística de la regla, Val(R)*
      - *Eliminar en R la última condición añadida, R^-*
      - *Mientras $Val(R^-)<Val(R)$, elimjinar condiciones de la regla*
  - *Seleccionar la mejor regla modificada, R'*
  - *Eliminar los ejemplos en E cubiertos por R'*
  - *Continuar*

- En este caso, las reglas dependen del orden de evaluación: lista de decisión

Ejemplo: lentes de contacto

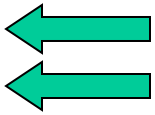
AGE	SPECTACLE	ASTIGMATISM	TEAR RATE	LENSEs
young	myope	no	reduced	none
young	myope	no	normal	soft
young	myope	yes	reduced	none
young	myope	yes	normal	hard
young	hypermetrope	no	reduced	none
young	hypermetrope	no	normal	soft
young	hypermetrope	yes	reduced	none
young	hypermetrope	yes	normal	hard
pre-presbyopic	myope	no	reduced	none
pre-presbyopic	myope	no	normal	soft
pre-presbyopic	myope	yes	reduced	none
pre-presbyopic	myope	yes	normal	hard
pre-presbyopic	hypermetrope	no	reduced	none
pre-presbyopic	hypermetrope	no	normal	soft
pre-presbyopic	hypermetrope	yes	reduced	none
pre-presbyopic	hypermetrope	yes	normal	none
presbyopic	myope	no	reduced	none
presbyopic	myope	no	normal	none
presbyopic	myope	yes	reduced	none
presbyopic	myope	yes	normal	hard
presbyopic	hypermetrope	no	reduced	none
presbyopic	hypermetrope	no	normal	soft
presbyopic	hypermetrope	yes	reduced	none
presbyopic	hypermetrope	yes	normal	none

Aprendizaje de reglas

Reglas con max. precisión (PRISM)

- Clase: *hard:* **si ? entonces rec. lenses=hard** (4 ejemplos de la clase en 24)
- Siete opciones

age=young	2/8
age=pre-presbyopic	1/8
age= presbyopic	1/8
spec=myope	3/12
spec=hypermetrope	1/12
astigmatism=yes	4/12
tear rate=normal	4/12



- Sig. término: **si (astig.=yes) y ? entonces lenses = hard** (cuatro ejemplos de la clase en 12)

Reglas con max. precisión (PRISM)

- Seis opciones para segundo término

age=young	2/4
age=pre-presbyopic	1/4
age= presbyopic	1/4
spec=myope	3/6
spec=hypermetrope	1/6
tear rate=normal	4/6

- Sig. término: **si (astig.=yes) y (tear rate=normal) y ? entonces lenses = hard)** (4 ejemplos en 6)
- Cinco opciones para tercer término

age=young	2/2
age=pre-presbyopic	1/2
age= presbyopic	1/2
spec=myope	3/3
spec=hypermetrope	1/6

Reglas con max. precisión (PRISM)

- Resultado:

  si (astig.=yes) y (tear rate=normal) y (spec=myope)
  (entonces lenses = hard) (3/3)

- Tomando el ejemplo restante:

  si (age=young) y (astig.=yes) y (tear rate=normal) entonces
  (lenses = hard) (1/1) (2/2 del conjunto original)

Reglas con max. precisión

```
Si astigmatism = yes and tear-prod-rate = normal
   and spectacle-prescrip = myope then HARD (3/3)
Si age = young and astigmatism = yes
   and tear-prod-rate = normal then HARD (1/1)

Si astigmatism = no and tear-prod-rate = normal (3/3)
   and spectacle-prescrip = hypermetrope then SOFT
Si astigmatism = no and tear-prod-rate = normal
   and age = young then SOFT (1/1)
Si age = pre-presbyopic and astigmatism = no
   and tear-prod-rate = normal then SOFT (1/1)

Si tear-prod-rate = reduced then NONE (12/12)
Si age = presbyopic and tear-prod-rate = normal
  and spectacle-prescrip = myope and astigmatism = no then NONE (1/1)
Si spectacle-prescrip = hypermetrope and astigmatism = yes
   and age = pre-presbyopic then NONE (1/1)
Si age = presbyopic and spectacle-prescrip = hypermetrope
   and astigmatism = yes then NONE (1/1)
```

Poda con evaluación (INDUCT)

- Mejor regla para valor de clase “hard”:

si (astig.=yes) ^ (tear rate=normal) ^ (spec=myope) entonces (lenses = hard) (3/3)

P=4, T=24, p=3, t=3, u=3

$$\mathrm{Val}(\mathrm{R}) = \mathrm{p}(3,3) = \frac{1}{\binom{24}{3}}\binom{4}{3}\binom{20}{0} = 0.0019$$

- Quitando la última condición:

si (astig.=yes) ^ (tear rate=normal) entonces (lenses = hard) (4/6)

P=4, T=24, p=4, t=6, u=4

$$\mathrm{Val}(\mathrm{R}^{-}) = \mathrm{p}(4,6) = \frac{1}{\binom{24}{6}}\binom{4}{4}\binom{20}{2} = 0.0014$$

Poda con evaluación (INDUCT)

- Quitando la penúltima condición:

si (astig.=yes) entonces (lenses = hard) (4/12)
P=4, T=24, p=4, t=12, u=4

$$\mathrm{Val}(\mathrm{R}^{-}) = \mathrm{p}(4,12) = \frac{1}{\binom{24}{12}}\binom{4}{4}\binom{20}{8} = 0.047$$

- Mejor regla para valor de clase “hard”:

si (astig.=yes) ^ (tear rate=normal) entonces (lenses = hard) (4/6)
val=0.0014

Poda con evaluación (INDUCT)

- Mejor regla para valor de clase “none”:

si (tear rate=reduced) entonces (lenses = none) (12/12)
P=15, T=24, p=12, t=12, u=12

$$\mathrm{Val}(\mathrm{R}^{-}) = \mathrm{p}(4,12) = \frac{1}{\binom{24}{12}}\binom{15}{12}\binom{9}{0} = 1.68\mathrm{e}-4$$

- **No hay poda**

Poda con evaluación (INDUCT)

Mejor regla para valor de clase “soft”:

si (astigmatism=no) and (tear-rate=normal) and (spec-pres=hypermetrope) entonces (lenses = soft) (3/3)

P=5, T=24, p=3, t=3

$$\mathrm{Val}(\mathrm{R}) = \mathrm{p}(3,3) = \frac{1}{\binom{24}{3}}\binom{5}{3}\binom{19}{0} = 0.0049$$

- última condición:

si (astigmatism=no) and (tear-rate=normal)

entonces (lenses = soft) (5/6)

P=5, T=24, p=5, t=6

$$\mathrm{Val}(\mathrm{R}^{-}) = \mathrm{p}(5,6) = \frac{1}{\binom{24}{6}}\binom{5}{5}\binom{19}{1} = 1.41\mathrm{e}-4$$

Poda con evaluación (INDUCT)

- Se selecciona la mejor regla de entre las tres clases:
  si (astigmatism=no) and (tear-rate=normal)
  entonces (lenses = soft) (5/6)
- Se eliminan ejemplos cubiertos y se sigue iterando...

- Reglas resultantes (lista de decisión):
  - si (astigmatism = no) and (tear-prod-rate=normal): **soft** (5.0/6.0)
  - tear-prod-rate = reduced: **none** (12.0/12.0)
  - spectacle-prescrip = myope: **hard** (3.0/3.0)
  - : **none** (2.0/3.0)

Otras cuestiones

- Ejemplos con faltas en atributos o atributos numéricos
  - Los ejemplos con faltas son utilizados de forma natural
  - Atributos numéricos: solución similar a C4.5
- Poda con conjunto de test independiente (*reduced error pruning)*
  - Alternativa a la medida probabilística del valor de la regla
  - Se divide el conjunto de entrenamiento en *crecimiento* y *test* (2/3:1/3)
  - Desventaja de aprendizaje con menos ejemplos
- Reglas con excepciones
  - Formulación más legible y fácil de adaptar incrementalmente
  - Selección de clase más numerosa: valor por defecto
  - Se invoca recursivamente hasta tener elementos de una sola clase

Reglas de Asociación

- Clasificación sobre cualquier atributo/conjunto de atributos
- No es un problema de clasificación, sino de búsqueda general de relaciones significativas (aprendizaje no supervisado)
- Ejemplo:
  1. clase=+ => A4=Libro 3 (3:3) conf:(1)
  2. clase=- => A3=1 (3:3) conf:(1)
  3. A2=2 => A4=Libro 3 (3:3) conf:(1)
  4. A4=Libro, clase=- => A3=1 (2:2) conf:(1)
  5. A2=2, clase=+ 2 => A4=Libro (2:2) conf:(1)
  6. A2=2, A3=1 2 => A4=Libro (2:2) conf:(1)
  7. A1=1, clase=+ 2 => A4=Libro (2:2) conf:(1)
  8. A1=1, A3=1 => clase=- (2:2) conf:(1)
  9. A1=1, clase=- => A3=1 (2:2) conf:(1)
  10. A2=0 => A1=1 (2:2) conf:(1)

Reglas de Asociación

– Algoritmo “A priori”

- Buscar grupos de condiciones con cobertura suficiente
  - Grupos de 1 elemento
  - Combinaciones de 2 elementos
  - Combinaciones de 3 elementos
  - ...
- Seleccionar reglas con mejor precisión en cada grupo

- Hipótesis de trabajo: para que un grupo de k condiciones tenga una cobertura superior a s, todos sus subconjuntos de k-1 deben tenerla también:
  - Se almacenan con tablas de Hash para comprobar que existen los subconjuntos