Arboles de Decisión (III)

Carlos Hurtado L.

Depto de Ciencias de la Computación, Universidad de Chile

¿Cuál es el Mejor Modelo?

- Dado un conjunto de datos D
- Distintos algoritmos de aprendizaje pueden generar distintos modelos a partir de D.

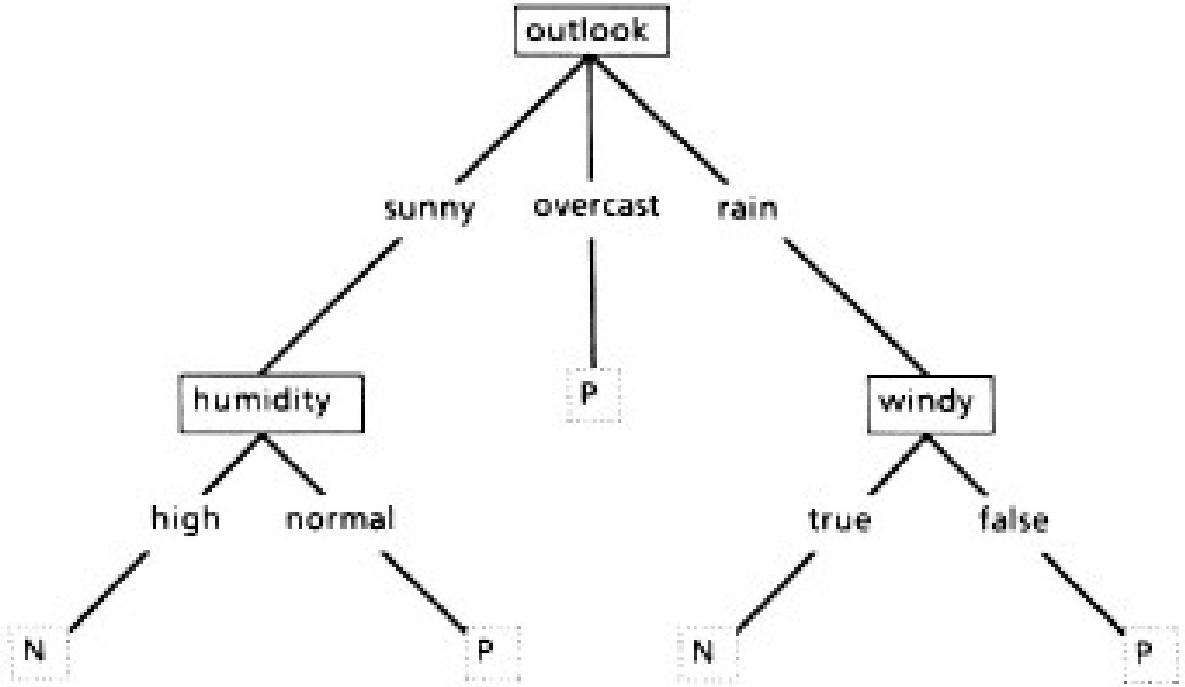
¿Cuál es el Mejor Modelo?

- Los mejores modelos son aquellos con:
  - menor error de predicción -- mejor poder predictivo.
  - menor complejidad
    - Principio de la “Navaja de Occam”: l*ex parsimoniae*: favorecer lo sucinto
    - Si tenemos dos árboles con el mismo error preferimos el menos complejo (más pequeño).

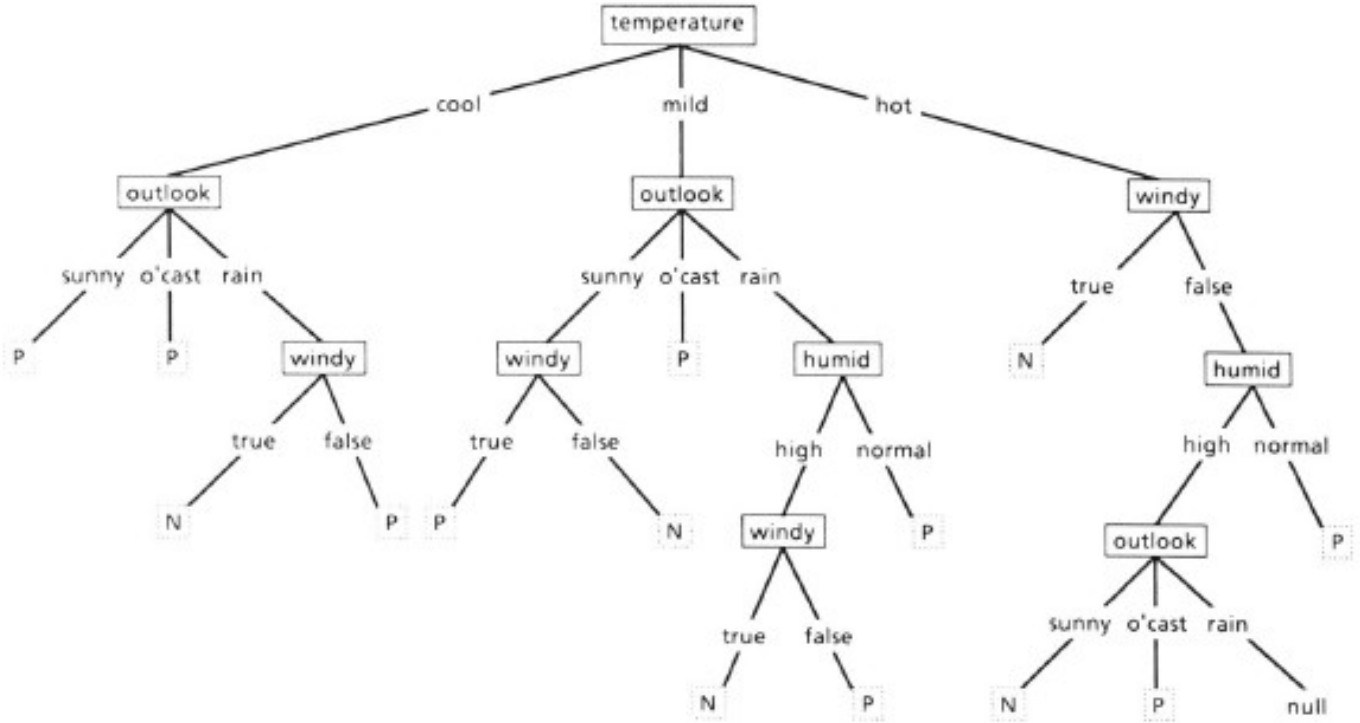
Ejemplo: Complejidad de un Modelo

Outlook	Temp.	Humidity	Windy	Play
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Rainy	Mild	High	FALSE	Yes
Rainy	Cool	Normal	FALSE	Yes
Rainy	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Sunny	Mild	High	FALSE	No
Sunny	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	FALSE	Yes
Sunny	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Mild	High	TRUE	No

Arbol Simple



Arbol Complejo



Error de Predicción

- Un modelo se induce de un conjunto de datos: **datos de entrenamiento**
- Una vez construido, el modelo se usa para predecir la clase de otros datos: **datos objetivos**
- **Error de predicción:** fracción de datos objetivos para los cuales se predijo mal la clase.

Error de Predicción

- En general no lo conocemos con exactitud, sólo podemos estimarlo
- Estrategia más común:
  - **Hold-out**: usamos un conjunto de datos independientes de los datos de entrenamiento para estimar el error: **datos de prueba.**

Estimación del Error de Predicción

- Error de predicción se estima usando técnicas de inferencia estadística
- **Inferencia estadística**: métodos para deducir propiedades de una **población** a partir de una pequeña parte de ella (**muestra**)
  - Por ejemplo, queremos predecir el resultado de una elección en base a una encuesta (muestra).

En el Contexto de Aprendizaje de Máquinas

- Datos de entrenamiento: datos que se usan para entrenar el modelo
- **Muestra**: Datos de prueba
- **Población**: Datos objetivo

Variables y Población

- En inf. estadística estudiamos como se comporta una variable X en la población.
- Para esto, la población se puede abstraer como un variable aleatoria X
  - Por ejemplo, podríamos estudiar la estatura de la población: $X \in [0, 180]$
  - O podríamos estudiar las preferencias en una elección de dos candidatos:

    $X \in \{0, 1\}$

En el Contexto de Aprendizaje de Máquinas

- La población son los datos objetivos.
- La variable a estudiar es una variable binaria $X \in \{0, 1\}$, donde $X = 0$ significa que se predice bien la clase y $X = 1$ que se predice mal.

Variable a Estudiar

- Variable aleatoria X con distribución de probabilidades $\Pr(X = x) = p(x)$
- Parámetros de la población:
  - Media: $\mu = E(X) = \sum xp(x)$
  - Varianza:

$$\sigma^2 = E((X - \mu)^2) = \sum (x - \mu)^2 p(x)$$

En el Contexto de Aprendizaje de Máquina

- La variable es binaria $X \in \{0, 1\}$
- La media de X la denotamos π y representa la proporción de la población para la cual $X = 1$
- La media π es el error de predicción (es una proporción).
- La varianza de X es $\pi(1 - \pi)$

Muestra Aleatoria Muy Simple (MAMS)

- Una muestra es un subconjunto de la población.
- El conjunto de variables aleatorias $\{X_1, \ldots, X_n\}$ representa una muestra de tamaño n de X
- MAMS: Los X_i son independientes y distribuyen como X
- Media de la muestra: $\bar{X} = \frac{1}{n} \sum_i X_i$

Propiedades de la Media de la Muestra

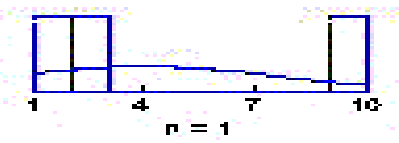
- Media de $\bar{X}$

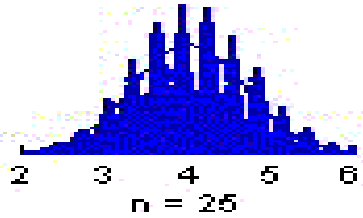
$$E(\bar{X}) = E(\frac{1}{n}\sum X_i) = \mu$$

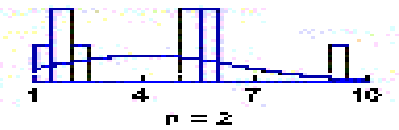
- Varianza de $\bar{X}$

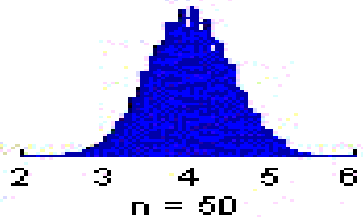
$$E((\bar{X} - \mu)^2) = var(\frac{1}{n}\sum X_i) = \frac{\sigma^2}{n}$$

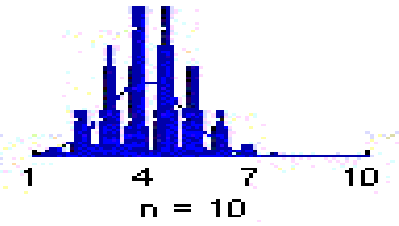
Distribución de la Media de Muestrapara distintos n

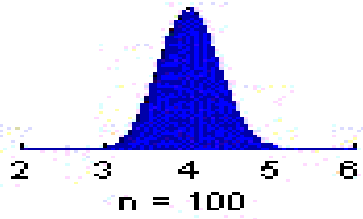












Teorema del Límite Central

- Si n es grande entonces la media de muestra distribuye aproximadamente normal.
- Esto se aplica independientemente de si la población distribuye normal.
- Es decir a medida que n aumenta, tenemos que

$$\bar{X} \sim N(\mu, \frac{\sigma}{n})$$

- Adicionalmente, a medida que n aumenta, la varianza disminuye -- tiende a cero.

En el Contexto de Aprendizaje de Máquina

- X es binaria ($X \in \{0, 1\}$)
- La media π es el error de predicción.
- La varianza de X es $\pi(1 - \pi)$
- Como $\bar{X}$ es una proporción, la denotamos p. Tenemos

$$p \sim N(\pi, \frac{\pi(1 - \pi)}{n}$$

Ejemplo

- ¿De nuestros 15 primeros nietos cuál es la chance que tengamos más de 10 hombre?
- En este caso sabemos que $\pi = 0.5$
- Luego

$$p \sim N(\pi, \frac{\pi(1-\pi)}{n})$$

$$p \sim N(0.5, 0.017)$$

Ejemplo (cont.)

- Tenemos $p \sim N(0.5, 0.129^2)$
- Queremos calcular $\Pr(p > \frac{10}{15})$
- Lo hacemos pasando a valor z y buscando la sgte probabilidad en una tabla de normal unitaria.

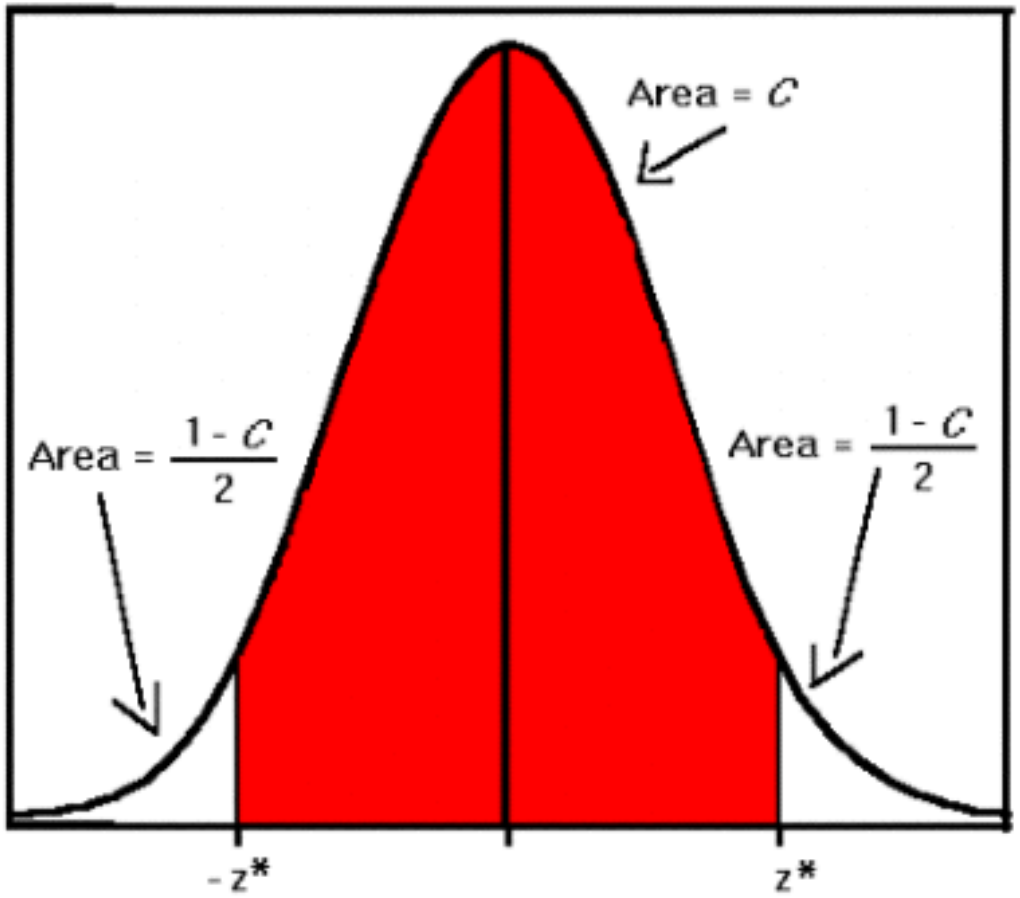
$$\Pr(\frac{p - 0.5}{0.129} > \frac{\frac{10}{15} - 0.5}{0.129}) = \Pr(Z > 1.29) = 0.099$$

Intervalos de Confianza para la Media de la Población

- Debido a que $\bar{X} \sim N(\mu, \frac{\sigma}{n})$
- Podemos definir un intervalo con C% confianza para μ

$$\mu = \bar{X} \pm z_{1-\frac{C}{2}} \frac{\sigma}{\sqrt{n}}$$

Nivel de confianza



Varianza Desconocida y pocos Datos

- Si no conocemos la varianza y n es pequeño (n < 100) en la práctica, se usa la sgte aproximación:

$$\frac{\bar{X} - \mu}{\frac{s}{\sqrt{n}}} \sim t(n-1) \qquad s = \frac{1}{n-1} \sum (X_i - \bar{X})^2$$

- Obtenemos el sgte intervalo con C% de confianza:

$$\mu = \bar{X} \pm t_{1-\frac{C}{2}} \frac{s}{\sqrt{n}}$$

En el Contexto de Aprendizaje de Máquinas

- Debido a que $p \sim N(\pi, \frac{\pi(1-\pi)}{n})$
- Podemos definir un intervalo con C% de confianza para el error

$$\pi = p \pm z_{1-\frac{C}{2}} \sqrt{\frac{\pi(1-\pi)}{n}}$$

Intervalo de Confianza para el Error de Predicción

$$\pi = p \pm z\sqrt{\frac{\pi(1-\pi)}{n}}$$

Despejamos π y obtenemos:

$$\pi = \frac{p + \frac{z^2}{2n} \pm z\sqrt{\frac{p}{n} - \frac{p^2}{n} + \frac{z^2}{4n}}}{1 + \frac{z^2}{n}}$$

Causas del Error de Predicción

- El error de predicción de un modelo se debe a tres causas:
  - **Varianza**: datos de entrenamiento no son representativos de datos objetivos.
  - **Sesgo:** modelo es limitado y aunque lo entrenemos con datos de prueba representativo igual tendremos error.
  - **Ruido:** Nuestro espacio de variables no separa adecuadamente las clases.

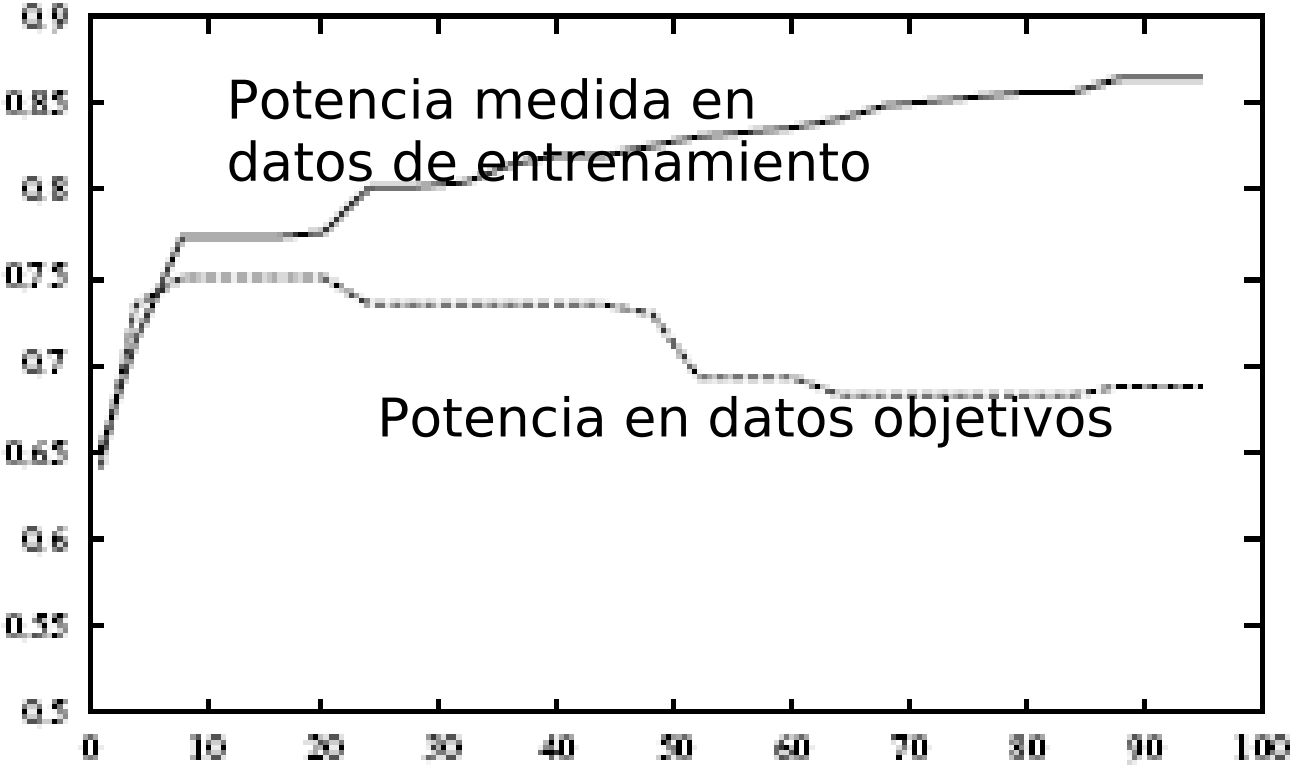
Error por Varianza

- Se produce por datos de entrenamiento no representativos
- Si tenemos menos datos es más alta la probabilidad de que no sean representativos
- Sin embargo, aunque tengamos muchos datos, estos también pueden no ser representativos

Error versus Complejidad del Modelo

- Si dejamos crecer mucho un árbol de decisión en el entrenamiento, puede aumentar el error por varianza: **sobreajuste**.
- Si no dejamos que crezca lo suficiente, puede aumentar el error por sesgo.

Error vs. Complejidad del Modelo



Sobreajuste

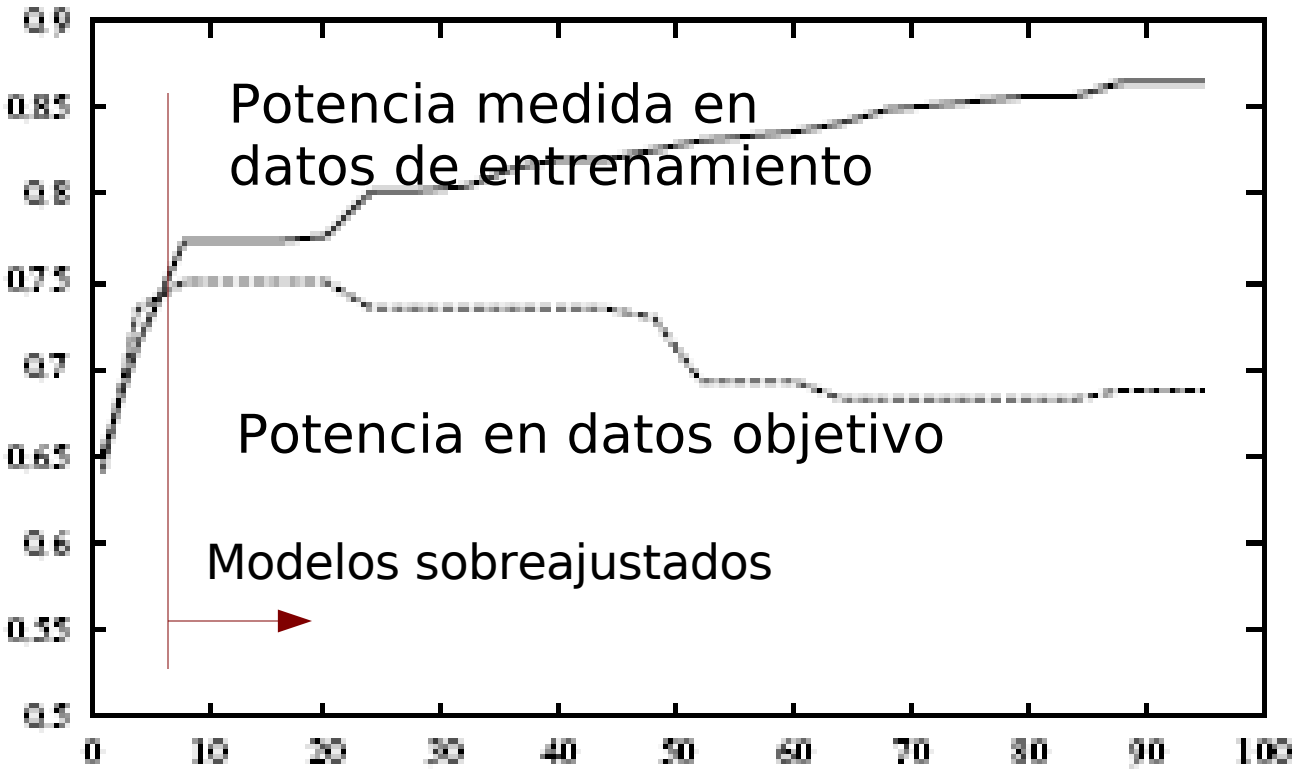
- El modelo está ajustado en exceso a los datos de entrenamiento.
- Se produce un error de predicción alto debido a varianza.
- El error que observamos en datos de entrenamiento es bajo pero el error en datos objetivos es alto.
- El modelo se generaliza mal a datos objetivos.

Sobreajuste: Definición

Dado un espacio de modelos M, un modelo m en M es sobreajustado si existe otro modelo m’ en M tal que:

- m tiene menor error que m’ en datos de entrenamiento y
- m’ tiene menor error que m en datos objetivo

Sobreajuste



Potencia
(1-Error)

Tamaño del árbol (número de nodos)

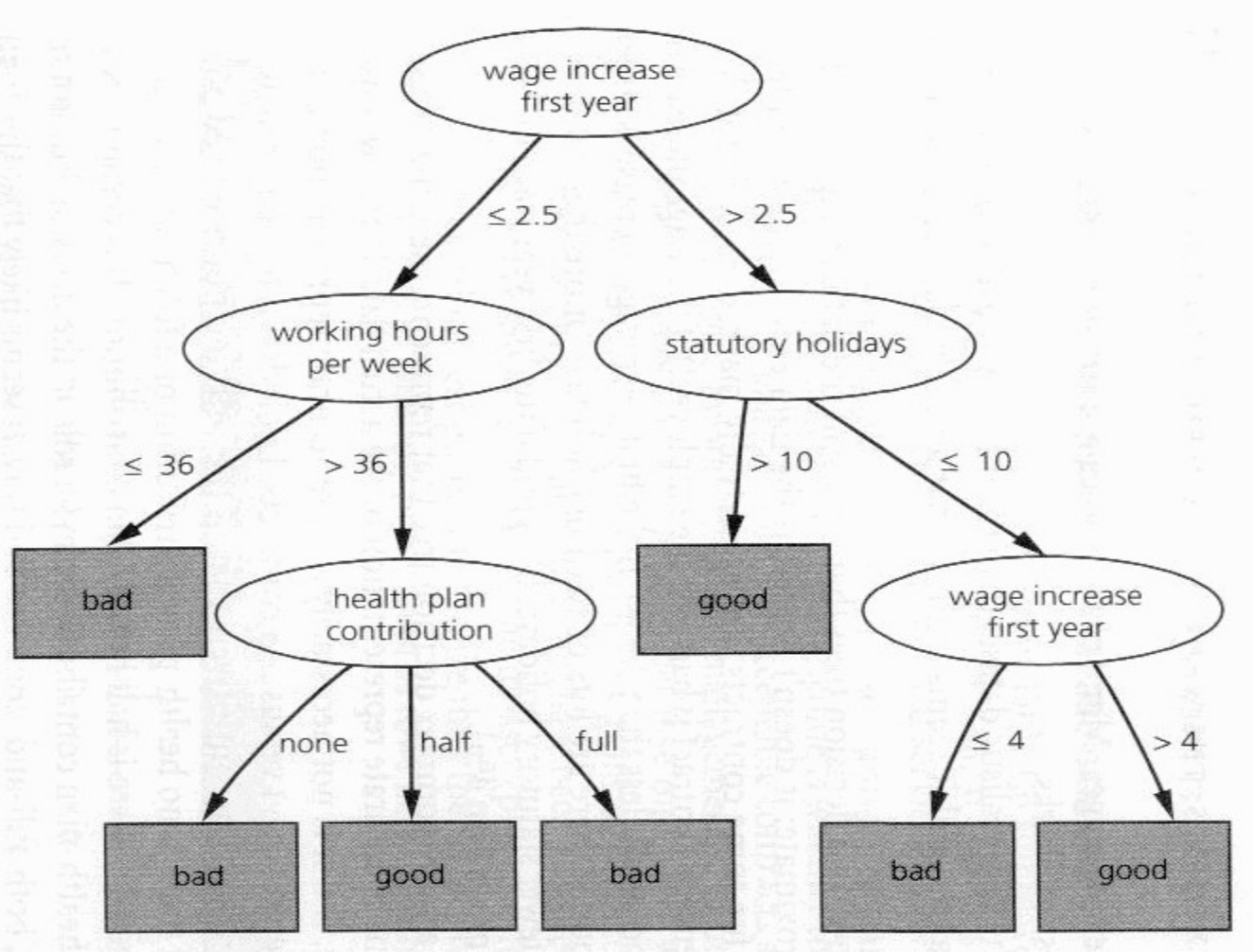
Ejemplo Sobreajuste

- Datos de resultados de negociaciones contractuales en Canadá, 1987-88
- Fuente: Weka
- Organizaciones con más de 500 miembros
  - profesores, enfermeras, policía, administrativos universitarios, etc.
- Cada contrato es clasificado bueno o malo para los empleados

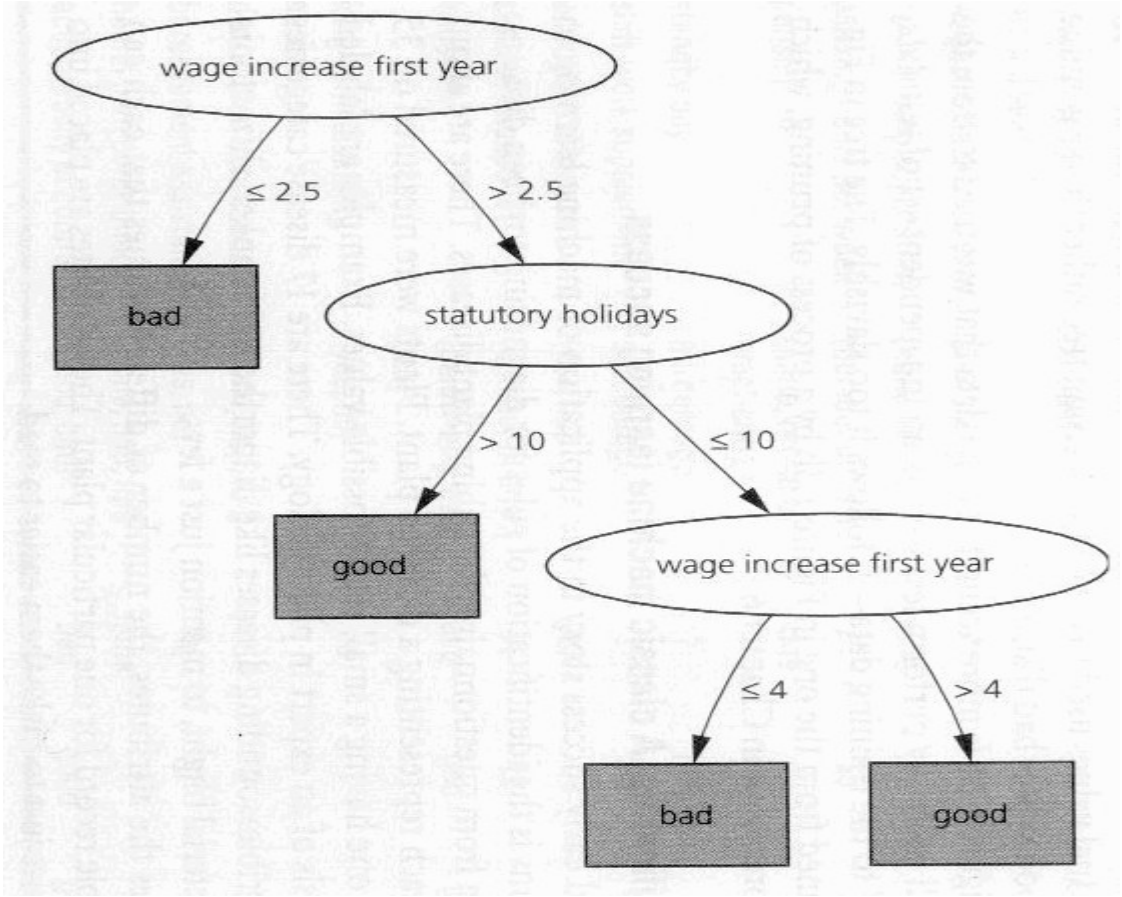
Table 1.6 The labor negotiations data.

attribute	type	1	2	3	...	40
duration	(number of years)	1	2	3		2
wage increase first year	percentage	2%	4%	4.3%		4.5
wage increase second year	percentage	?	5%	4.4%		4.0
wage increase third year	percentage	?	?	?		?
cost of living adjustment	{none, tcf, tc}	none	tcf	?		none
working hours per week	(number of hours)	28	35	38		40
pension	{none, ret-allw, empl-cntr}	none	?	?		?
standby pay	percentage	?	13%	?		?
shift-work supplement	percentage	?	5%	4%		4
education allowance	{yes, no}	yes	?	?		?
statutory holidays	(number of days)	11	15	12		12
vacation	{below-avg, avg, gen}	avg	gen	gen		avg
long-term disability assistance	{yes, no}	no	?	?		yes
dental plan contribution	{none, half, full}	none	?	full		full
bereavement assistance	{yes, no}	no	?	?		yes
health plan contribution	{none, half, full}	none	?	full		half

Arbol Sobreajustado



Arbol con Menor Error de Predicción



Poda

- Mecanismo para obtener árboles con menor error de predicción
- Pre-poda:
  - Parar la construcción del árbol en algunas nodos
- Post-poda
  - Construir un árbol complejo (posiblemente sobreajustado) y podarlo después.

Pre-poda

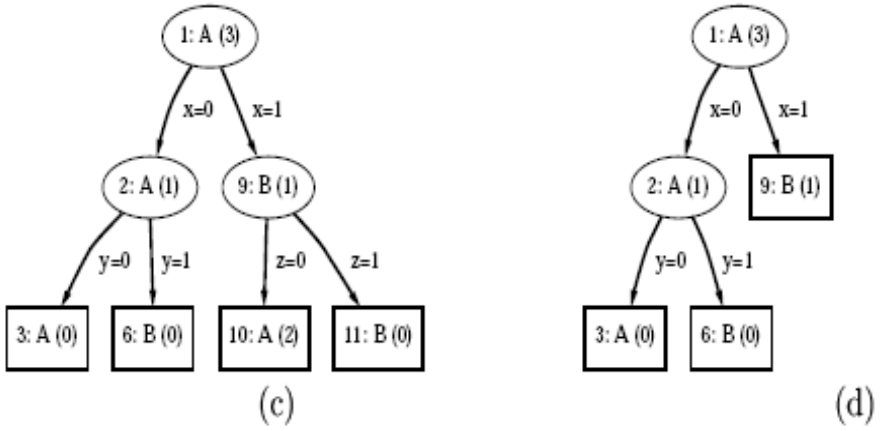
- Parar la construcción del árbol en algunos nodos en base a criterios como:
  - No expandir si *GiniSplit* < *k*
  - No expandir si el nodo tiene menos de *k* datos
  - No expandir si test de chi-cuadrado no rechaza independencia
- Al no expandir, etiquetar el nodo con la clase más frecuente en los datos asociados al nodo.

Post-poda

1. Basada en reducción del error
   - Podamos si el error en datos objetivos estimado del árbol resultante es menor.
   - Variante: primero transformamos el árbol en un conjunto de reglas y luego podamos.
2. Basada en Principio de Descripción Mínima
   - Modela compromiso error vs. complejidad
   - la idea es que un buen modelo minimiza el número de bits necesarios para codificar el modelo y los datos usando el modelo.

Post-poda Basada en Reducción del Error

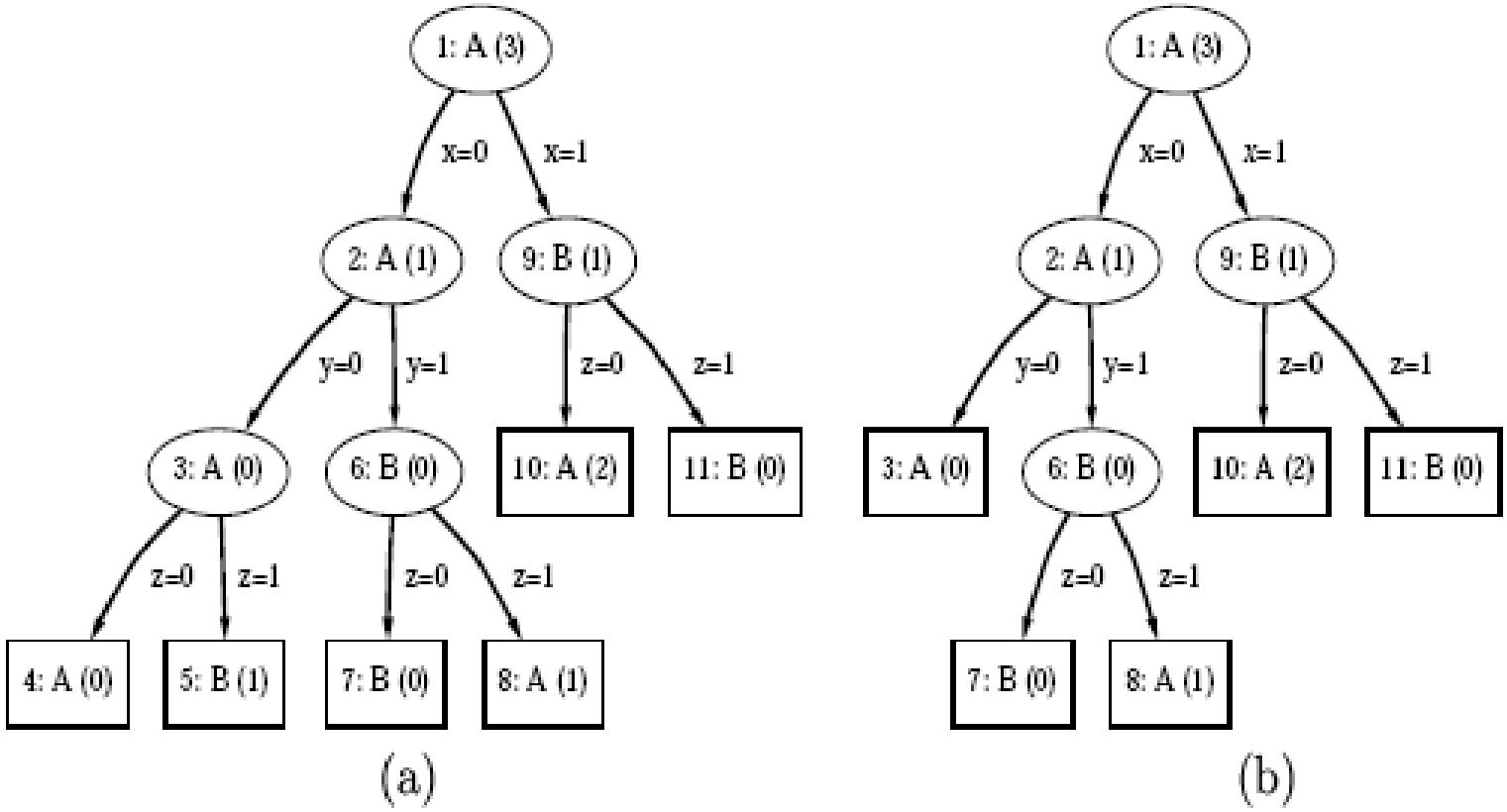
- Existen varias operaciones de poda.
- La operación más común es reemplazar el subárbol completo bajo un nodo por una hoja.



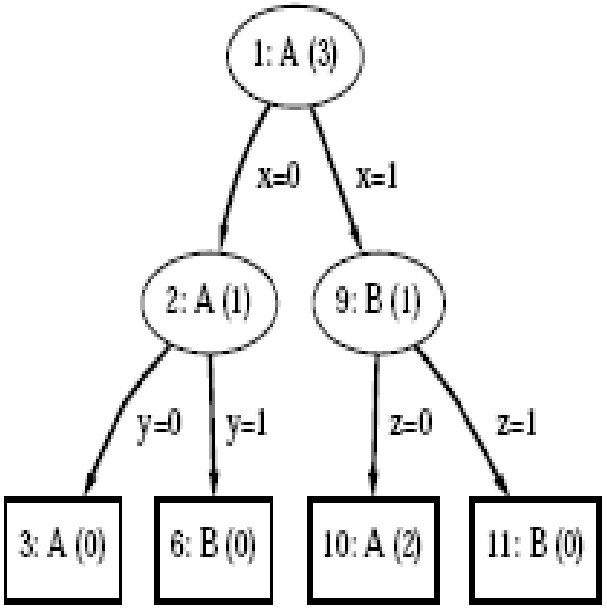
Algoritmo de Poda Basada en Reducción del Error

- La operación de poda consiste en reemplazar un subárbol por una hoja y etiquetar la hoja con la clase mayoritaria.
- Se recorre el árbol desde las hojas hacia arriba podando nodos.
- Para cada nodo que visitamos en el recorrido aplicamos un test para decidir si lo podamos o no.

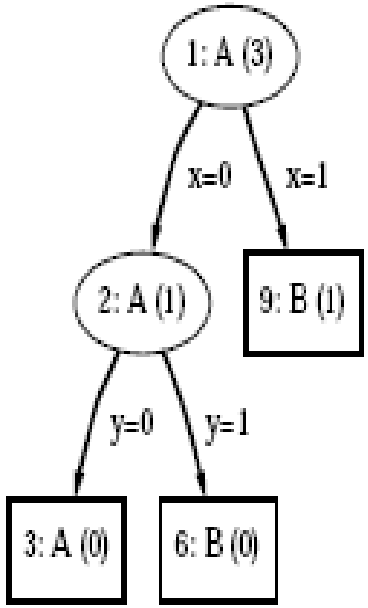
Ejemplo


1: A (3)
x=0
x=1
2: A (1)
9: B (1)
y=0
y=1
z=0
z=1
3: A (0)
6: B (0)
10: A (2)
11: B (0)
z=0
z=1
z=0
z=1
4: A (0)
5: B (1)
7: B (0)
8: A (1)
(a)
1: A (3)
x=0
x=1
2: A (1)
9: B (1)
y=0
y=1
z=0
z=1
3: A (0)
6: B (0)
10: A (2)
11: B (0)
z=0
z=1
7: B (0)
8: A (1)
(b)

Ejemplo


1: A (3)
x=0
x=1
2: A (1)
9: B (1)
y=0
y=1
z=0
z=1
3: A (0)
6: B (0)
10: A (2)
11: B (0)

(c)


1: A (3)
x=0
x=1
2: A (1)
9: B (1)
y=0
y=1
3: A (0)
6: B (0)

(d)

Test para decidir si podamos un nodo

- A = estimación error si no se poda el nodo
  - Estimamos el error del subárbol bajo el nodo.
  - Igual a la suma ponderada de errores estimados en nodos hijos
    - Si nodos hijos son hojas:
      - su error se estima usando inferencia estadística (intervalo de confianza)
    - Si nodo hijo no es hoja:
      - su error se estima como la suma ponderada de los nietos...
      - Se sigue sucesivamente hasta llegar a nodos hojas.
- B = estimación error si se poda el nodo
    - Se estima usando inferencia estadística.
- Si B < A se poda el nodo.

Test para decidir si podamos un nodo

- En general, el error de un nodo hoja se estima como el límite superior de un intervalo de confianza para el error objetivo.
- La muestra de datos para calcular este intervalo se puede obtener de:
  - los mismos datos de entrenamiento, o
  - muestra disjunta de datos de entrenamiento: **datos de poda**

Ejemplo: Poda en Algoritmo C4.5

- Se implementa post-poda basado en reducción del error.
- Se usa un nivel de confianza bajo: C=50%, lo que da z=0.69
- El error se estima como el límite superior del intervalo de confianza.

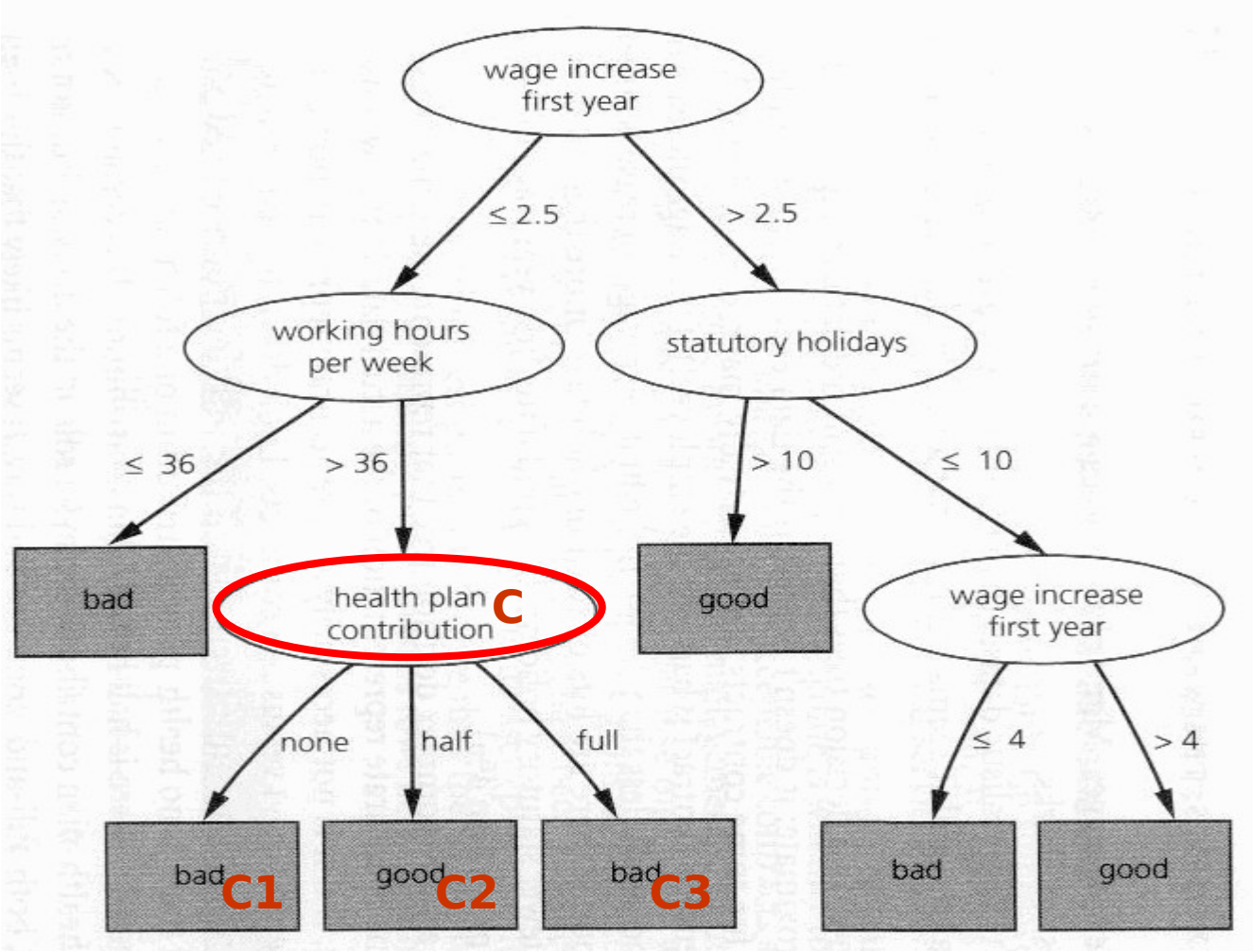
Ejemplo: Poda en algoritmo C4.5

- Para un nodo hoja el error se estima como:

$$\pi = \frac{p + \frac{z^2}{2n} + z\sqrt{\frac{p}{n} - \frac{p^2}{n} + \frac{z^2}{4n}}}{1 + \frac{z^2}{n}}$$

- donde $z = 0.69$, p es la proporción de errores medidos y n es el número de datos asociados al nodo

Ejemplo: Poda en C4.5



Ejemplo: Tenemos los Siguientes Datos de Poda (DP)

WI	WH	HP	Clase
2	38	none	Good
1	40	none	Good
1.5	40	none	Bad
2	40	none	Bad
2.2	38	none	Bad
1	40	none	Bad
1.5	40	half	Good
2	38	half	Bad
2	38	full	Good
1	40	full	Good
1	40	full	Bad
1	38	full	Bad
2	38	full	Bad
2	38	full	Bad

Ejemplo: Poda en C4.5

- Corremos el modelo sobre DP y obtenemos las siguientes frecuencias:

Nodo	Clase	Num Good	Num Bad	Errores
c1	Bad	2	4	2
c2	Good	1	1	1
c3	Bad	2	4	2

Ejemplo: Error del Nodo c si no Podamos

- Para cada nodo hoja estimamos como el límite superior del intervalo para el error objetivo con nivel de confianza 50% (z=0.69).

Nodo	Num. Errores	Num Datos	p	Error Est.
c1	2	6	0.33	0.47
c2	1	2	0.5	0.72
c3	2	6	0.33	0.47

El error estimado del nodo c si no podamos es: (6/14) .47 + (2/14) .72 + (6/14) .47 = .51

Ejemplo: Error del Nodo c si Podamos

- Asumamos que si podamos la clase mayoritaria observada en los datos de entrenamientos es “Bad”
- Luego tenemos los siguientes:

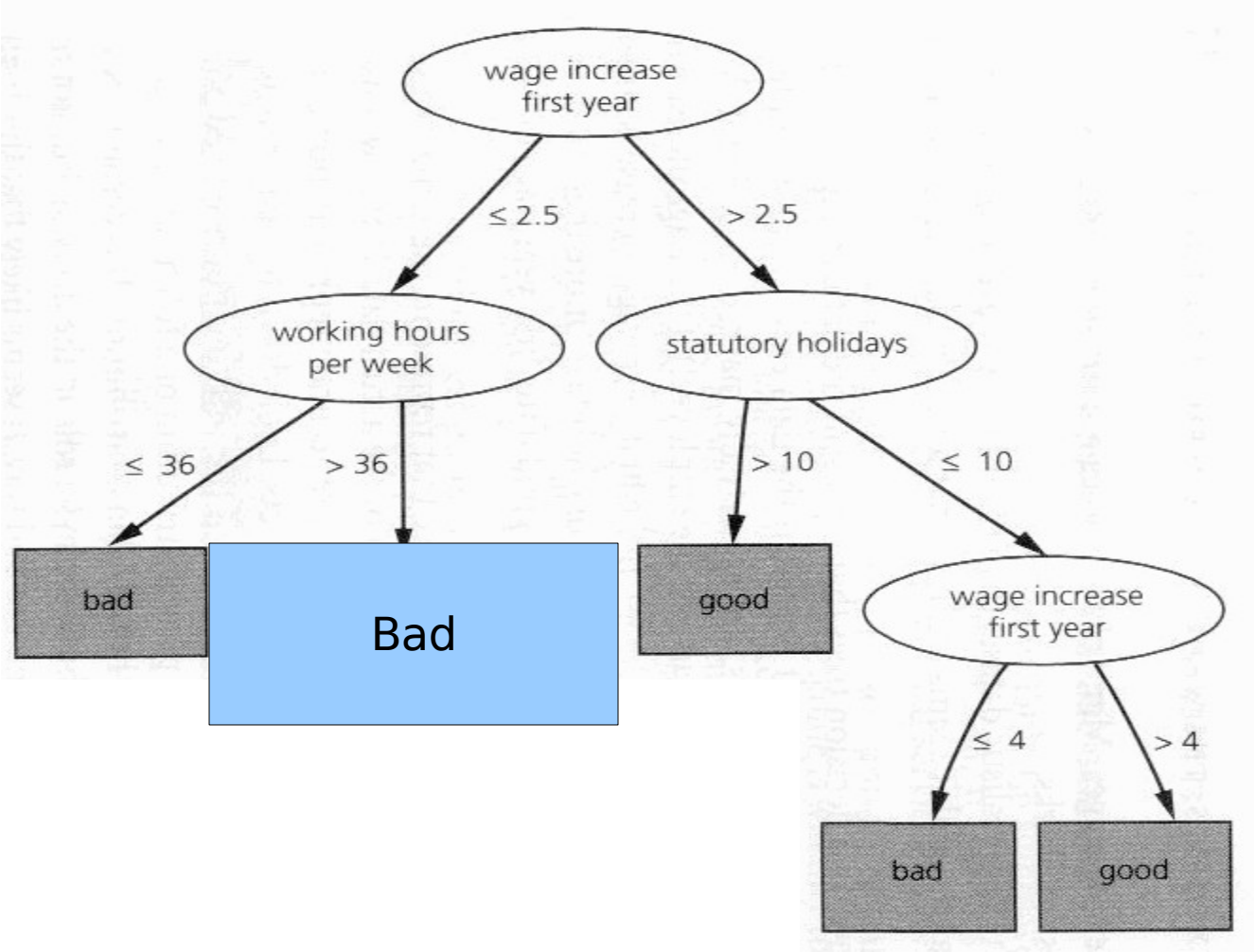
Nodo	Clase	Num Good	Num Bad	Errores
c1	Bad	2	4	2
c2	Good	1	1	1
c3	Bad	2	4	2
c	Bad	5	9	5

- Para el nodo c tenemos n= 14, p= 5/14, luego error estimado = 0.46

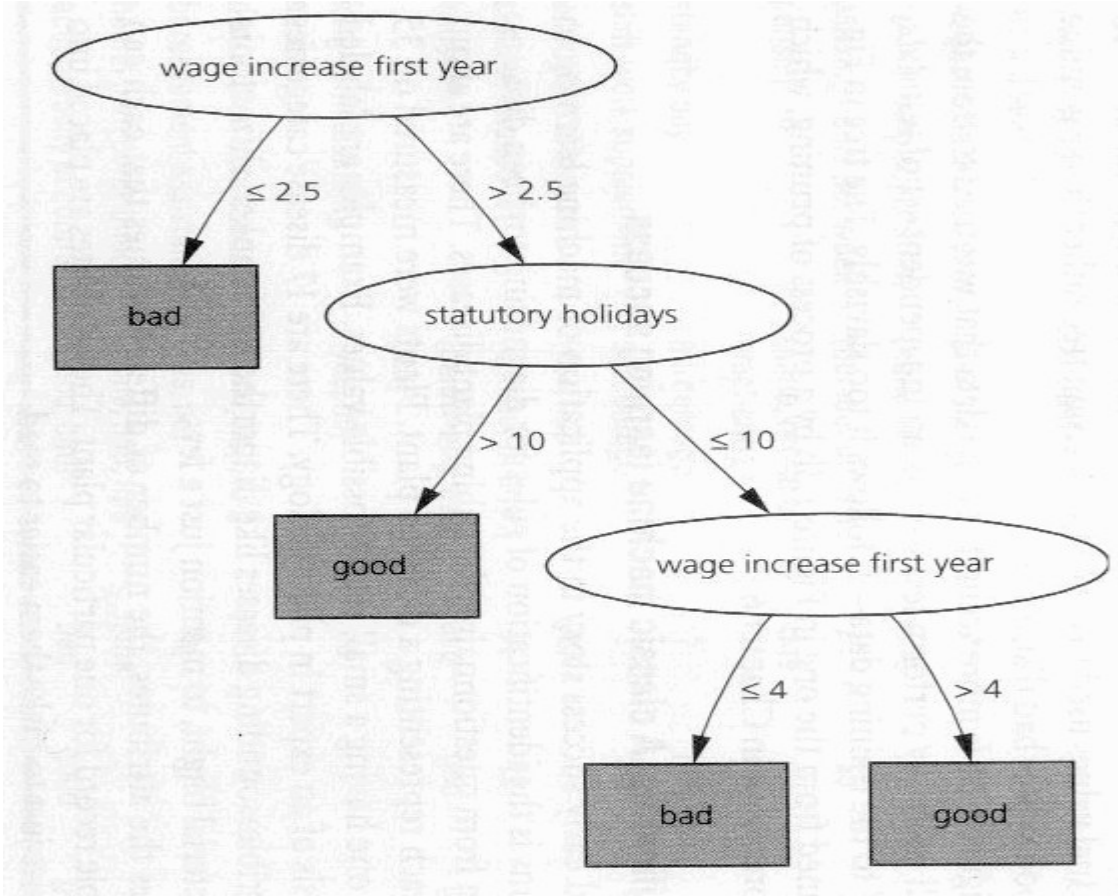
Ejemplo: poda en C4.5

- Como 0.51 > 0.46, podamos el nodo.
- El nodo c se transforma en una hoja
- Etiquetamos c con la clase “Bad”

Ejemplo: árbol resultante de primera poda



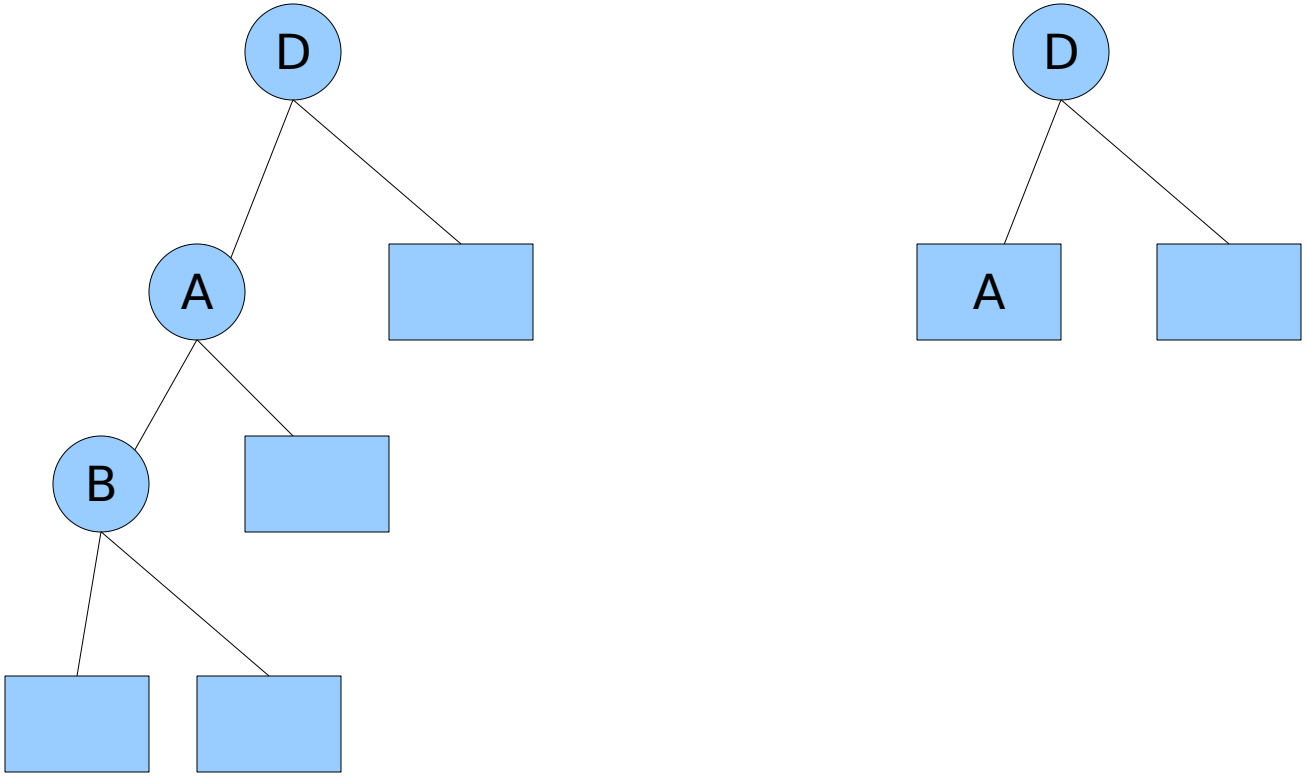
Ejemplo: árbol final



Algoritmo de Poda Basada en Reducción del Error

- Recorremos el árbol podando de abajo hacia arriba.
- El podar el árbol de esta forma garantiza que obtenemos el subárbol de menor error.
- Ejercicio: dar un ejemplo de un árbol que al no podarlo de abajo hacia arriba no generamos el árbol con menor error.

Ejemplo: Podamos A antes de B



Ejemplo: Podamos B antes de A