Introducción al Aprendizaje automático

Dra Ana Georgina Flesia

Optativa Ciencias de la Computación
FaMAF-UNC
Oficina 370
georgina.flesia@unc.edu.ar

2020

Árboles de Decisión: ejemplo

Atributos y clase:

tamaño	color	forma	clase
pequeño	rojo	círculo	$+$
grande	azul	cuadrado	$-$
	verde	triángulo	

Árboles de Decisión: ejemplo

Atributos y clase:

tamaño	color	forma	clase
pequeño	rojo	círculo	$+$
grande	azul	cuadrado	$-$
	verde	triángulo	

Instancias:

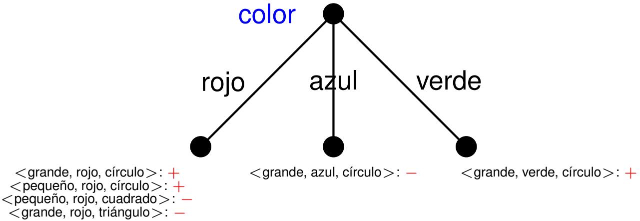
$$\begin{array}{rc} \langle\text{grande, rojo, círculo}\rangle: & + \\ \langle\text{pequeño, rojo, círculo}\rangle: & + \\ \langle\text{pequeño, rojo, cuadrado}\rangle: & - \\ \langle\text{grande, azul, círculo}\rangle: & - \\ \langle\text{grande, verde, círculo}\rangle: & + \\ \langle\text{grande, rojo, triángulo}\rangle: & - \end{array}$$

Árboles de Decisión

Construcción de un árbol a partir de los ejemplos: Los nodos evalúan características, con una rama para cada posible valor de la característica, y se continúa hasta que las hojas especifican la categoría:

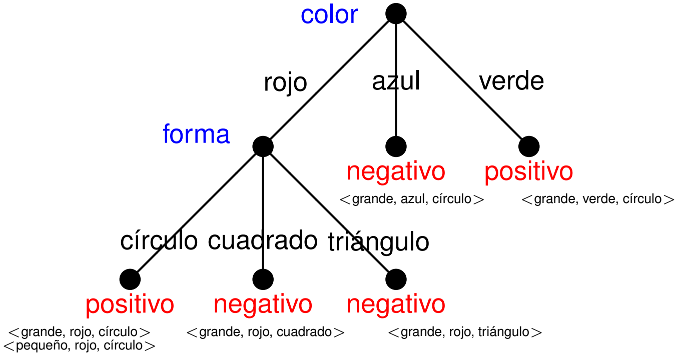
Árboles de Decisión

Construcción de un árbol a partir de los ejemplos: Los nodos evalúan características, con una rama para cada posible valor de la característica, y se continúa hasta que las hojas especifican la categoría:



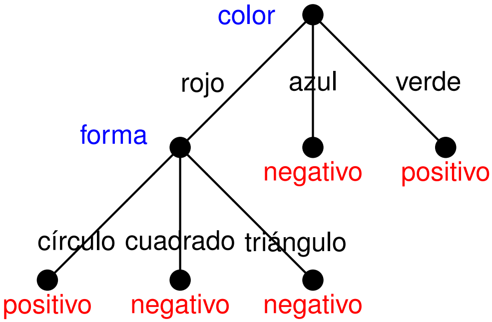
Árboles de Decisión

Construcción de un árbol a partir de los ejemplos: Los nodos evalúan características, con una rama para cada posible valor de la característica, y se continúa hasta que las hojas especifican la categoría:



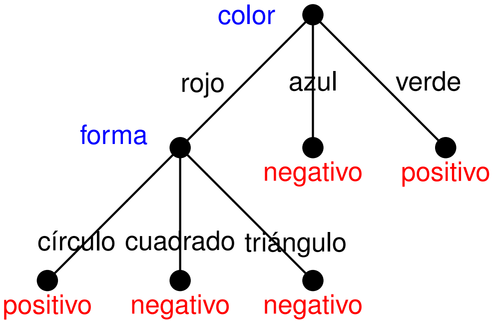
Árboles de Decisión

Los nodos evalúan características, con una rama para cada posible valor de la característica, y las hojas especifican la categoría



Árboles de Decisión

Los nodos evalúan características, con una rama para cada posible valor de la característica, y las hojas especifican la categoría



El árbol construído puede codificarse como una cascada `if, then, else`

Árboles de Decisión

Ante una nueva instancia no etiquetada:

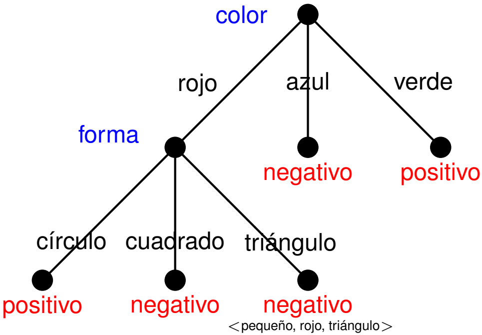
$<$pequeño, rojo, triángulo$>$

Árboles de Decisión

Ante una nueva instancia no etiquetada:

<pequeño, rojo, triángulo>

el árbol construído funciona como clasificador:



Particularidades de los Árboles de Decisión

Particularidades de los Árboles de Decisión

- Las características con valores contínuos se pueden partir en dos o más rangos, mediante un umbral (p.e. longitud < 3 y longitud ≥ 3)

Particularidades de los Árboles de Decisión

- Las características con valores contínuos se pueden partir en dos o más rangos, mediante un umbral (p.e. longitud < 3 y longitud ≥ 3)
- Existen métodos para tratar datos faltantes

Particularidades de los Árboles de Decisión

- Las características con valores contínuos se pueden partir en dos o más rangos, mediante un umbral (p.e. longitud < 3 y longitud ≥ 3)
- Existen métodos para tratar datos faltantes
- Los árboles de clasificación tienen etiquetas de clase discretas en las hojas, mientras que los árboles de regresión tienen valores contínuos

Particularidades de los Árboles de Decisión

- Las características con valores contínuos se pueden partir en dos o más rangos, mediante un umbral (p.e. longitud < 3 y longitud ≥ 3)
- Existen métodos para tratar datos faltantes
- Los árboles de clasificación tienen etiquetas de clase discretas en las hojas, mientras que los árboles de regresión tienen valores contínuos
- Los algoritmos para construir árboles son eficientes para el procesamiento de grandes cantidades de datos

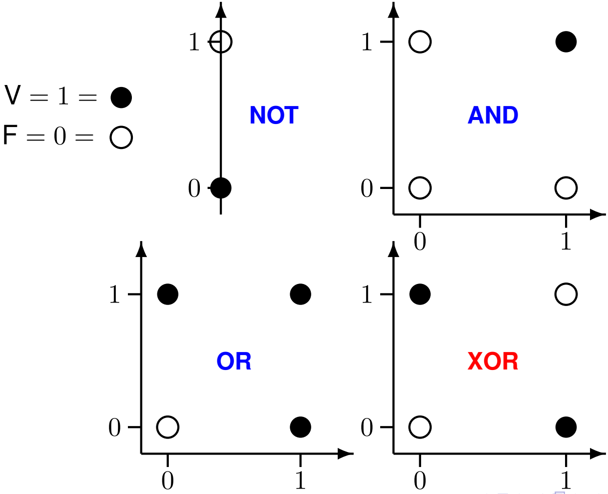
Particularidades de los Árboles de Decisión

- Las características con valores contínuos se pueden partir en dos o más rangos, mediante un umbral (p.e. longitud < 3 y longitud ≥ 3)
- Existen métodos para tratar datos faltantes
- Los árboles de clasificación tienen etiquetas de clase discretas en las hojas, mientras que los árboles de regresión tienen valores contínuos
- Los algoritmos para construir árboles son eficientes para el procesamiento de grandes cantidades de datos
- Existen métodos para tratar datos de entrenamiento ruidosos, con errores tanto en en las características como en la clase

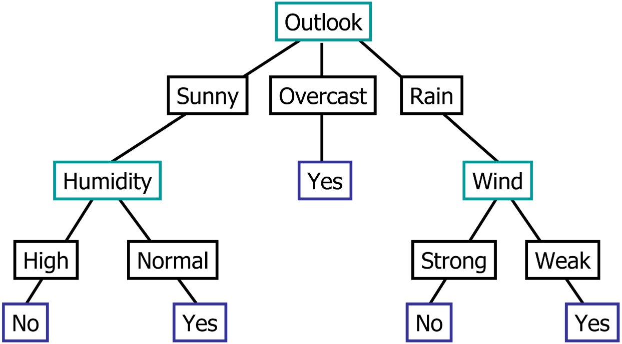
Particularidades de los Árboles de Decisión

- Las características con valores contínuos se pueden partir en dos o más rangos, mediante un umbral (p.e. longitud < 3 y longitud ≥ 3)
- Existen métodos para tratar datos faltantes
- Los árboles de clasificación tienen etiquetas de clase discretas en las hojas, mientras que los árboles de regresión tienen valores contínuos
- Los algoritmos para construir árboles son eficientes para el procesamiento de grandes cantidades de datos
- Existen métodos para tratar datos de entrenamiento ruidosos, con errores tanto en en las características como en la clase
- Los árboles pueden representar cualquier función de clasificación

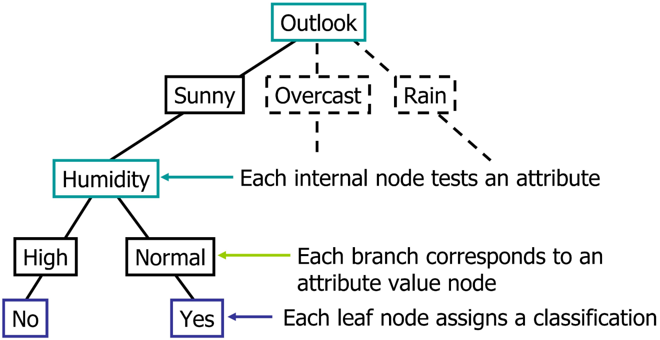
Árboles de Decisión



Árbol de Decisión para PlayTennis

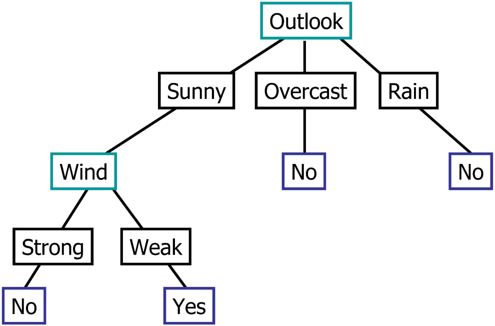


Árbol de Decisión para PlayTennis



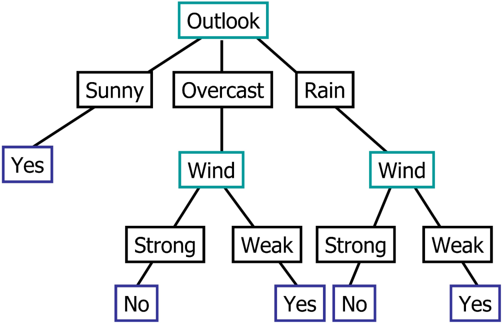
Árbol de Decisión para Conjunción

Outlook=Sunny $\wedge$ Wind=Weak

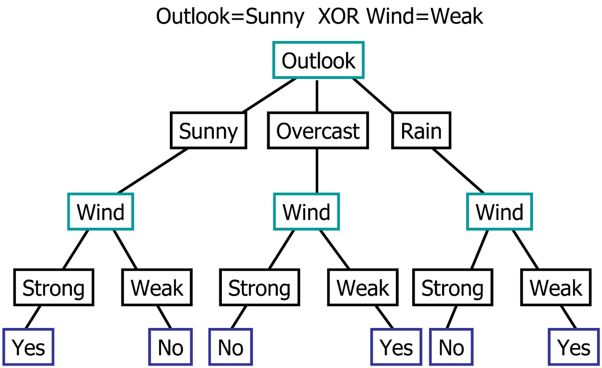


Árbol de Decisión para Disyunción

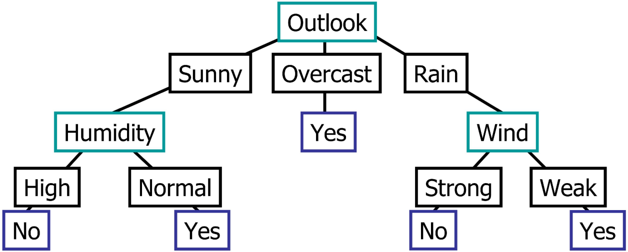
Outlook=Sunny ∨ Wind=Weak



Árbol de Decisión para XOR



Árboles de decisión representan disyunciones de conjunciones



$$
\begin{aligned}
& (\text{Outlook}=\text{Sunny} \wedge \text{Humidity}=\text{Normal}) \\
\vee\ & (\text{Outlook}=\text{Overcast}) \\
\vee\ & (\text{Outlook}=\text{Rain} \wedge \text{Wind}=\text{Weak})
\end{aligned}
$$

Elección de la característica en un nodo

Elección de la característica en un nodo

- El objetivo es obtener el árbol más chico posible (MDL)

Elección de la característica en un nodo

- El objetivo es obtener el árbol más chico posible (MDL)
- El método recién empleado (top-down) hace una búsqueda voraz (greedy), por lo cual no garantiza encontrar el árbol más chico posible, si bien en general encuentra una buena solución

Elección de la característica en un nodo

- El objetivo es obtener el árbol más chico posible (MDL)
- El método recién empleado (top-down) hace una búsqueda voraz (greedy), por lo cual no garantiza encontrar el árbol más chico posible, si bien en general encuentra una buena solución
- Se elige la característica que crea subconjuntos de ejemplos relativamente "puros" en una sola clase, de forma que las hojas queden más cerca de la raíz

Elección de la característica en un nodo

- El objetivo es obtener el árbol más chico posible (MDL)
- El método recién empleado (top-down) hace una búsqueda voraz (greedy), por lo cual no garantiza encontrar el árbol más chico posible, si bien en general encuentra una buena solución
- Se elige la característica que crea subconjuntos de ejemplos relativamente “puros” en una sola clase, de forma que las hojas queden más cerca de la raíz
- Hay muchas heurísticas para elegir una característica. La más popular se basa en Ganancia de Información (Information Gain) propuesta por Quinlan (1979)

Entropía de Shannon

Entropía de Shannon

- La entropía de un conjunto de ejemplos S, relativo a una clasificación binaria (0 y 1) es

$$\text{Entropy}(S) = -p_0 \log_2(p_0) - p_1 log_2(p_1)$$

donde p_1 es la fracción de ejemplos positivos en S y $p_0 = 1 - p_1$ es la fracción de negativos

Entropía de Shannon

- La entropía de un conjunto de ejemplos S, relativo a una clasificación binaria (0 y 1) es

$$\text{Entropy}(S) = -p_0 \log_2(p_0) - p_1 log_2(p_1)$$

  donde p_1 es la fracción de ejemplos positivos en S y $p_0 = 1 - p_1$ es la fracción de negativos
- Si todos los ejemplos están en una categoría, la entropía es 0

Entropía de Shannon

- La entropía de un conjunto de ejemplos S, relativo a una clasificación binaria (0 y 1) es

$$\text{Entropy}(S) = -p_0 \log_2(p_0) - p_1 log_2(p_1)$$

  donde p_1 es la fracción de ejemplos positivos en S y $p_0 = 1 - p_1$ es la fracción de negativos
- Si todos los ejemplos están en una categoría, la entropía es 0
- Si los ejemplos están mezclados en partes iguales ($p_1 = p_0 = 0.5$), la entropía alcanza su máximo en 1

Entropía de Shannon

- La entropía de un conjunto de ejemplos S, relativo a una clasificación binaria (0 y 1) es

$$\text{Entropy}(S) = -p_0 \log_2(p_0) - p_1 log_2(p_1)$$

  donde p_1 es la fracción de ejemplos positivos en S y $p_0 = 1 - p_1$ es la fracción de negativos
- Si todos los ejemplos están en una categoría, la entropía es 0
- Si los ejemplos están mezclados en partes iguales ($p_1 = p_0 = 0.5$), la entropía alcanza su máximo en 1
- La entropía representa el número medio de bits que se necesitan para codificar la clase en S

Entropía de Shannon

- La entropía de un conjunto de ejemplos S, relativo a una clasificación binaria (0 y 1) es

$$\text{Entropy}(S) = -p_0 \log_2(p_0) - p_1 log_2(p_1)$$

  donde p_1 es la fracción de ejemplos positivos en S y $p_0 = 1 - p_1$ es la fracción de negativos
- Si todos los ejemplos están en una categoría, la entropía es 0
- Si los ejemplos están mezclados en partes iguales ($p_1 = p_0 = 0.5$), la entropía alcanza su máximo en 1
- La entropía representa el número medio de bits que se necesitan para codificar la clase en S
- Para problemas multi-clase con c categorías, la entropía se generaliza según

$$\text{Entropy}(S) = -\sum_{i=1}^{c} p_i \log_2(p_i)$$

Ganancia de Información

- La ganancia de información de un set de ejemplos respecto de una característica F es la información mutua que resulta al dividir según esta característica

$$\text{Gain}(S, F) = \text{Entropy}(S) - \sum_{v=\text{values(F)}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

donde S_v es el subconjunto de S que tiene valor v para la característica F

Ganancia de Información

- La ganancia de información de un set de ejemplos respecto de una característica F es la información mutua que resulta al dividir según esta característica

$$\text{Gain}(S, F) = \text{Entropy}(S) - \sum_{v=\text{values(F)}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

  donde S_v es el subconjunto de S que tiene valor v para la característica F
- La entropía de cada subconjunto resultante está ponderado por su tamaño (cantidad de elementos que contiene)

Ejemplo de Ganancia de Información:

S:

e_1 =<grande, rojo, círculo>: $+$ e_2 =<pequeño, rojo, círculo>: $+$

e_3 =<pequeño, rojo, cuadrado>: $-$ e_4 =<grande, azul, círculo>: $-$

$$\text{Entropy}(S) = -2(\frac{1}{2}\log_2(\frac{1}{2}))$$

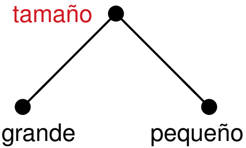
Ejemplo de Ganancia de Información:

S:

$e_1 =$<grande, rojo, círculo>: $+$ $e_2 =$<pequeño, rojo, círculo>: $+$

$e_3 =$<pequeño, rojo, cuadrado>: $-$ $e_4 =$<grande, azul, círculo>: $-$

$$\text{Entropy}(S) = -2(\frac{1}{2}\log_2(\frac{1}{2}))$$



e_1+, e_4- e_2+, e_3-

$E_G = 1$ $E_P = 1$

$$G = 1 - (0.5 \times 1 + 0.5 \times 1) = 0$$

$$\text{Entropy}(S_G) = -2(\frac{1}{2}\log_2(\frac{1}{2})) = 1$$

$$\text{Entropy}(S_P) = -2(\frac{1}{2}\log_2(\frac{1}{2})) = 1$$

$$\text{Gain} = \text{Entropy}(S) - \frac{|S_G|}{|S|}\text{Entropy}(S_G) - \frac{|S_P|}{|S|}\text{Entropy}(S_P)$$

$$= 1 - \frac{2}{4}1 - \frac{2}{4}1 = 0$$

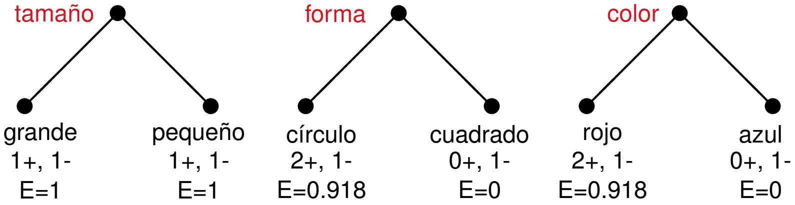
Ejemplo de Ganancia de Información:

S: <grande, rojo, círculo>: $+$ <pequeño, rojo, círculo>: $+$
<pequeño, rojo, cuadrado>: $-$ <grande, azul, círculo>: $-$

Ejemplo de Ganancia de Información:

S: <grande, rojo, círculo>: $+$ <pequeño, rojo, círculo>: $+$

<pequeño, rojo, cuadrado>: $-$ <grande, azul, círculo>: $-$



$G = 1 - (0.5 \times 1$
$+0.5 \times 1) = 0$

$G = 1 - (0.75 \times 0.918$
$+0.25 \times 0) = 0.311$

$G = 1 - (0.75 \times 0.918$
$+0.25 \times 0) = 0.311$

Uso de las características

Uso de las características

- Las características (no necesariamente todas) aparecen sólo una vez en los nodos (no se repiten)

Uso de las características

- Las características (no necesariamente todas) aparecen sólo una vez en los nodos (no se repiten)
- Una característica con valores contínuos, puede aparecer en más de un nodo pero con diferentes valores de corte

Uso de las características

- Las características (no necesariamente todas) aparecen sólo una vez en los nodos (no se repiten)
- Una característica con valores contínuos, puede aparecer en más de un nodo pero con diferentes valores de corte
- Ejemplo: Deporte al aire libre

Temperatura (^{o}C)	5	12	18	22	25	33
Práctica	no	no	sí	sí	sí	no

Búsqueda en el Espacio de Hipótesis

Búsqueda en el Espacio de Hipótesis

- Se trata de aprendizaje en batch, ya que los ejemplos de entrenamiento se procesan todos juntos, en contraste con un aprendizaje incremental que actualizaría la hipótesis después de cada ejemplo

Búsqueda en el Espacio de Hipótesis

- Se trata de aprendizaje en batch, ya que los ejemplos de entrenamiento se procesan todos juntos, en contraste con un aprendizaje incremental que actualizaría la hipótesis después de cada ejemplo
- Aplica búsqueda voraz que puede quedar limitada a una solución óptima local

Búsqueda en el Espacio de Hipótesis

- Se trata de aprendizaje en batch, ya que los ejemplos de entrenamiento se procesan todos juntos, en contraste con un aprendizaje incremental que actualizaría la hipótesis después de cada ejemplo
- Aplica búsqueda voraz que puede quedar limitada a una solución óptima local
- Se encuentra un árbol consistente con un conjunto de entrenamiento sin conflictos (de clase), pero no necesariamente el más simple

Búsqueda en el Espacio de Hipótesis

- Se trata de aprendizaje en batch, ya que los ejemplos de entrenamiento se procesan todos juntos, en contraste con un aprendizaje incremental que actualizaría la hipótesis después de cada ejemplo
- Aplica búsqueda voraz que puede quedar limitada a una solución óptima local
- Se encuentra un árbol consistente con un conjunto de entrenamiento sin conflictos (de clase), pero no necesariamente el más simple
- La Ganancia de Información tiene sesgo hacia los árboles poco profundos

Complejidad computacional

- Supongamos n ejemplos y m características

Complejidad computacional

- Supongamos n ejemplos y m características
- En el peor caso se tiene un árbol donde para alcanzar las hojas se tienen que evaluar todas las características

Complejidad computacional

- Supongamos n ejemplos y m características
- En el peor caso se tiene un árbol donde para alcanzar las hojas se tienen que evaluar todas las características
- Al nivel i se evaluan las $(m-i)$ características restantes y para calcular la ganancia de información se usan todos los ejemplos:

$$\sum_{i=1}^{m}(m-i)\,n \sim O(nm^2)$$

Complejidad computacional

- Supongamos n ejemplos y m características
- En el peor caso se tiene un árbol donde para alcanzar las hojas se tienen que evaluar todas las características
- Al nivel i se evaluan las $(m-i)$ características restantes y para calcular la ganancia de información se usan todos los ejemplos:

$$\sum_{i=1}^{m} (m-i)\, n \sim O(nm^2)$$

- En la práctica rara vez el árbol será completo y la complejidad usualmente resulta lineal en m y en n

Problema de sobreajuste (overfitting)

- Ocurre al aprender a clasificar perfectamente los datos de entrenamiento pero a costa de fallar en la tarea de generalizar

Problema de sobreajuste (overfitting)

- Ocurre al aprender a clasificar perfectamente los datos de entrenamiento pero a costa de fallar en la tarea de generalizar
- Potencia los problemas:
  - Ruido en los datos de entrenamiento
  - El algoritmo puede tomar decisiones basadas en datos que no reflejen la distribución de una mayor cantidad de ejemplos

Problema de sobreajuste (overfitting)

- Ocurre al aprender a clasificar perfectamente los datos de entrenamiento pero a costa de fallar en la tarea de generalizar
- Potencia los problemas:
  - Ruido en los datos de entrenamiento
  - El algoritmo puede tomar decisiones basadas en datos que no reflejen la distribución de una mayor cantidad de ejemplos
- Decimos que un clasificador sobreajusta si su performance es muy buena en el set de entrenamiento, pero en comparación desmejora mucho sobre un set de evaluación independiente

Ejemplo de Overffiting

Determinación experimental de la Ley de Ohm

$$I = 1/R\, V$$

Con un polinómio de grado nueve se puede ajustar *perfectamente* (sin error) los datos experimentales !

Ejemplo de Overffiting:

Determinación experimental de la Ley de Ohm

$$I = 1/R\ V$$

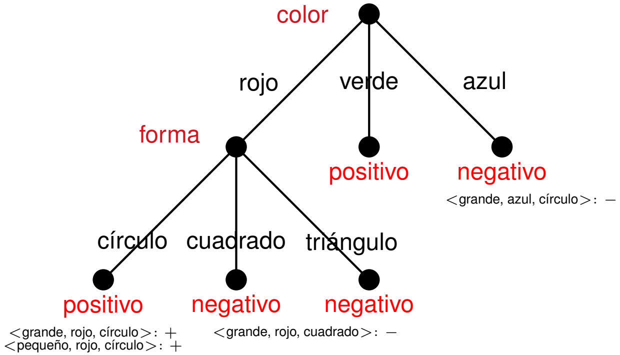
El ajuste lineal, si bien es menos preciso en el ajuste de los datos muestrales pero logra mayor generalización con los datos no usados en el ajuste

Sobreajuste de ruido

- El ruido en la categoría o la característica puede causar sobreajuste.
  Por ejemplo añadir la instancia ruidosa:
  $<$mediano, azul, círculo$>$: $+$ (que debe ser $-$!)

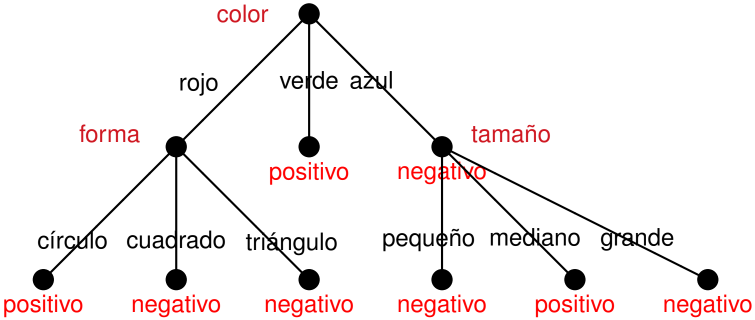
Sobreajuste de ruido

- El ruido en la categoría o la característica puede causar sobreajuste. Por ejemplo añadir la instancia ruidosa: $<$mediano, azul, círculo$>$: $+$ (que debe ser $-$!)



Sobreajuste de ruido

- El ruido en la categoría o la característica puede causar sobreajuste. Por ejemplo añadir la instancia ruidosa: $<$mediano, azul, círculo$>$: $+$ (que debe ser $-$!)



Prevención del sobreajuste: Pruning (Poda)

- Dos métodos básicos:

Prevención del sobreajuste: Pruning (Poda)

- Dos métodos básicos:
  - Prepruning: Detener en algún momento el crecimiento del árbol durante la construcción top-down cuando ya no hay suficientes datos para hacer decisiones criteriosas
    (por ejemplo tener un mínimo número de ejemplo por hoja)

Prevención del sobreajuste: Pruning (Poda)

- Dos métodos básicos:
  - Prepruning: Detener en algún momento el crecimiento del árbol durante la construcción top-down cuando ya no hay suficientes datos para hacer decisiones criteriosas
    (por ejemplo tener un mínimo número de ejemplo por hoja)
  - Postpruning: Luego de obtener el árbol completo, eliminar subárboles que no contienen suficiente evidencia.

Prevención del sobreajuste: Pruning (Poda)

- Dos métodos básicos:
  - Prepruning: Detener en algún momento el crecimiento del árbol durante la construcción top-down cuando ya no hay suficientes datos para hacer decisiones criteriosas
    (por ejemplo tener un mínimo número de ejemplo por hoja)
  - Postpruning: Luego de obtener el árbol completo, eliminar subárboles que no contienen suficiente evidencia.
- Luego de la poda, rotular la hoja resultante con la clase mayoritaria

Métodos para determinar qué ramas podar

Métodos para determinar qué ramas podar

- Validation: Reservar algunos datos de entrenamiento como conjunto de validación (validation set, tuning set) para evaluar si el error de clasificación postpruning no es peor que el anterior:
  reduced error-pruning method

Métodos para determinar qué ramas podar

- Validation: Reservar algunos datos de entrenamiento como conjunto de validación (validation set, tuning set) para evaluar si el error de clasificación postpruning no es peor que el anterior:
  reduced error-pruning method
- Evaluación estadística: usando los datos de entrenamiento implementar un test χ^2 para determinar si hay o no mejora de performance al retener una rama

Métodos para determinar qué ramas podar

- Validation: Reservar algunos datos de entrenamiento como conjunto de validación (validation set, tuning set) para evaluar si el error de clasificación postpruning no es peor que el anterior:
  reduced error-pruning method
- Evaluación estadística: usando los datos de entrenamiento implementar un test χ^2 para determinar si hay o no mejora de performance al retener una rama
- Mínima longitud de descripción (MDL): Determinar si la complejidad adicional de la hipótesis es menos compleja que simplemente recordar explícitamente todas las excepciones que resultan de la poda

Problemas usuales con la poda

Problemas usuales con la poda

- La evaluación estadística con los mismos datos de entrenamiento es poco confiable

Problemas usuales con la poda

- La evaluación estadística con los mismos datos de entrenamiento es poco confiable
- El problema de la validación es que potencialmente “gasta” datos de entrenamiento en el conjunto de validación

Problemas usuales con la poda

- La evaluación estadística con los mismos datos de entrenamiento es poco confiable
- El problema de la validación es que potencialmente “gasta” datos de entrenamiento en el conjunto de validación
- La severidad de este problema depende de dónde nos encontramos en la curva de aprendizaje:

Validación cruzada

- Uso de una métrica MDL

Validación cruzada

- Uso de una métrica MDL
- Se realizan pruebas de reduced error-pruning usando diferentes particiones aleatorias de los datos para obtener los conjuntos de aprendizaje y validación (usualmente 10-fold cross-validation)

Validación cruzada

- Uso de una métrica MDL
- Se realizan pruebas de reduced error-pruning usando diferentes particiones aleatorias de los datos para obtener los conjuntos de aprendizaje y validación (usualmente 10-fold cross-validation)
- Registrar la complejidad del árbol podado en cada fold de aprendizaje. Sea C el promedio de las complejidades medidas

Validación cruzada

- Uso de una métrica MDL
- Se realizan pruebas de reduced error-pruning usando diferentes particiones aleatorias de los datos para obtener los conjuntos de aprendizaje y validación (usualmente 10-fold cross-validation)
- Registrar la complejidad del árbol podado en cada fold de aprendizaje. Sea C el promedio de las complejidades medidas
- Construir un árbol final a partir de todos los datos de entrenamiento y detener la construcción al alcanzar la complejidad C

Validación cruzada

- Uso de una métrica MDL
- Se realizan pruebas de reduced error-pruning usando diferentes particiones aleatorias de los datos para obtener los conjuntos de aprendizaje y validación (usualmente 10-fold cross-validation)
- Registrar la complejidad del árbol podado en cada fold de aprendizaje. Sea C el promedio de las complejidades medidas
- Construir un árbol final a partir de todos los datos de entrenamiento y detener la construcción al alcanzar la complejidad C
- No hay pérdida de datos de entrenamiento

Saga de algoritmos

Saga de algoritmos

- Algoritmo ID3 (Quinlan 1986) utiliza Ganancia de Información

Saga de algoritmos

- Algoritmo ID3 (Quinlan 1986) utiliza Ganancia de Información
- Algoritmo C4.5 (Quinlan 1993)
  - incorpora pruning
  - maneja atributos con diferentes costes
  - maneja características con datos faltantes

Saga de algoritmos

- Algoritmo ID3 (Quinlan 1986) utiliza Ganancia de Información
- Algoritmo C4.5 (Quinlan 1993)
  - incorpora pruning
  - maneja atributos con diferentes costes
  - maneja características con datos faltantes
  - maneja características con datos discretos y contínuos

Saga de algoritmos

- Algoritmo ID3 (Quinlan 1986) utiliza Ganancia de Información
- Algoritmo C4.5 (Quinlan 1993)
  - incorpora pruning
  - maneja atributos con diferentes costes
  - maneja características con datos faltantes
  - maneja características con datos discretos y contínuos
  - J48 una implementación java open source del C4.5 en WEKA

Saga de algoritmos

- Algoritmo ID3 (Quinlan 1986) utiliza Ganancia de Información
- Algoritmo C4.5 (Quinlan 1993)
  - incorpora pruning
  - maneja atributos con diferentes costes
  - maneja características con datos faltantes
  - maneja características con datos discretos y contínuos
  - J48 una implementación java open source del C4.5 en WEKA
  - El costo numérico de computar logaritmos se puede solucionar usando la impureza de Gini

$$\text{Gini}(S) = \sum_{i=1}^{c} p_i \left(1 - p_i\right)$$

Saga de algoritmos

Saga de algoritmos

- Costes en los errores de clasificación $\rightarrow$ C5.0

Saga de algoritmos

- Costes en los errores de clasificación $\rightarrow$ C5.0
- Clase con valores contínuos (árboles de regresión) $\rightarrow$ CART

Saga de algoritmos

- Costes en los errores de clasificación $\rightarrow$ C5.0
- Clase con valores contínuos (árboles de regresión) $\rightarrow$ CART
- Aprendizaje Incremental $\rightarrow$ ID4 ID5 ID5R ID6MDL