Introducción a las técnicas de clasificación

Maria-Amparo Vila
vila@decsai.ugr.es

Grupo de Investigación en Bases de Datos y Sistemas de Información Inteligentes https://idbis.ugr.es/
Departamento de Ciencias de la Computación e Inteligencia Artificial
Universidad de Granada

Introducción

Concepto Básicos

Definición

Clasificación *es el proceso de aprender una función que aplica un conjunto de atributos* $X_1..X_n$ *en otro atributo* Y*. Si:*

- *Si* Y *es discreta, booleana, nominal etc. tenemos Modelos de clasificación propiamente dichos*
- *Si* Y *es continua tenemos Modelos de regresión*

La funcion que se aprende se denomina también Modelo de clasificación en general

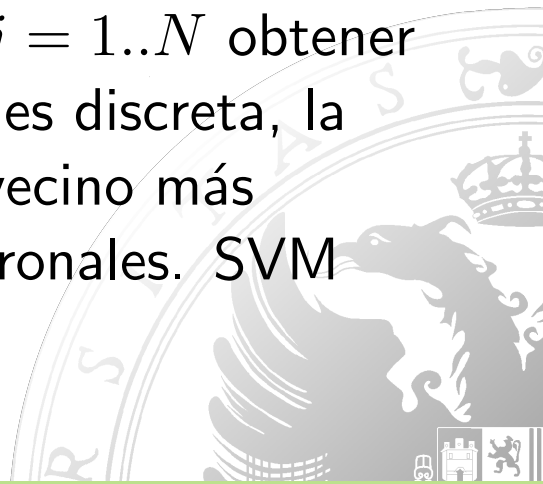
Introducción

Concepto Básicos

Según el objetivo del aprendizaje tenemos:

Modelos explicativos También llamados descriptivos. Intentan mostar cómo depende Y de $X_1..X_N$: árboles de decisión, clasificadores bayesianos, modelos regresión, modelos de reglas

Modelos predictivos . No buscan tanto mostrar la dependencia como dado un item o_i con valores $x_{ij}, j = 1..N$ obtener el valor y_i de la variable objetivo. Si Y es discreta, la clase a la que pertenece. Métodos del vecino más cercano, métodos basados en redes neuronales. SVM



Introducción

El proceso general de clasificación

1.- Se considera un data set con valores en Y.**Conjunto de entrenamiento**

items\variables	X_1	X_2	$\cdots\cdots$	X_N	Y
o_1	x_{11}	x_{12}	$\cdots\cdots$	x_{1N}	y_1
$\vdots$	$\vdots$	$\vdots$	$\cdots\cdots$	$\vdots$	$\vdots$
o_M	x_{M1}	x_{M2}	$\cdots\cdots$	x_{MN}	y_M

2.- Se construye (aprende) el modelos de clasificación

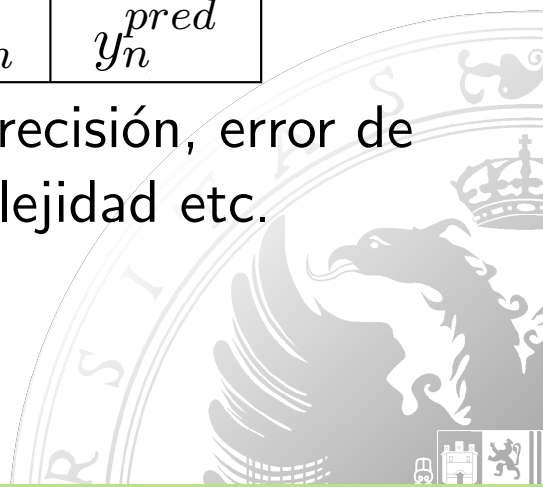
Introducción

El proceso general de clasificación

3.- Se prueba en otro dataset distinto **Conjunto test** caculando los valores de Y^{pred}

items\variables	X_1	$\cdots\cdots$	X_N	Y	Y^{pred}
o_1	x_{11}	$\cdots\cdots$	x_{1N}	y_1	y_1^{pred}
$\vdots$	$\vdots$	$\cdots\cdots$	$\vdots$	$\vdots$	$\vdots$
o_n	x_{n1}	$\cdots\cdots$	x_{nN}	y_n	y_n^{pred}

4.- Se evalúa en el modelo según distintos criterios: precisión, error de clasificación, escalabilidad, interpretabilidad, complejidad etc.



Introducción

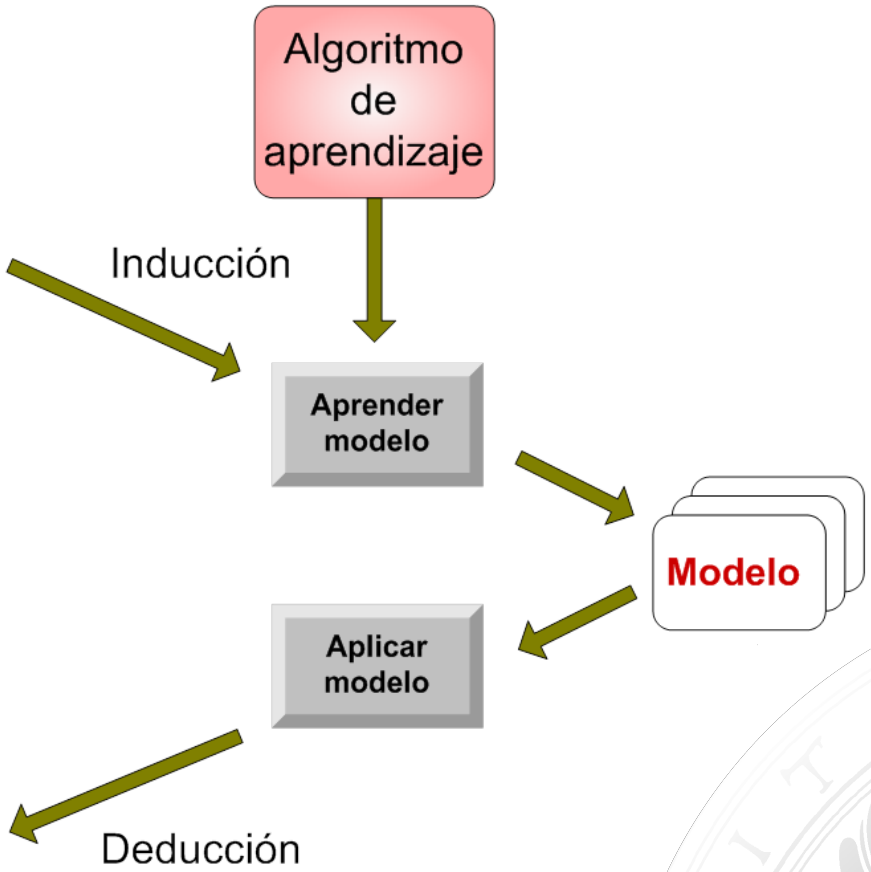
El proceso general de clasificación

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Conjunto de entrenamiento

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Conjunto de prueba



Clasificación mediante Arboles de Decisión

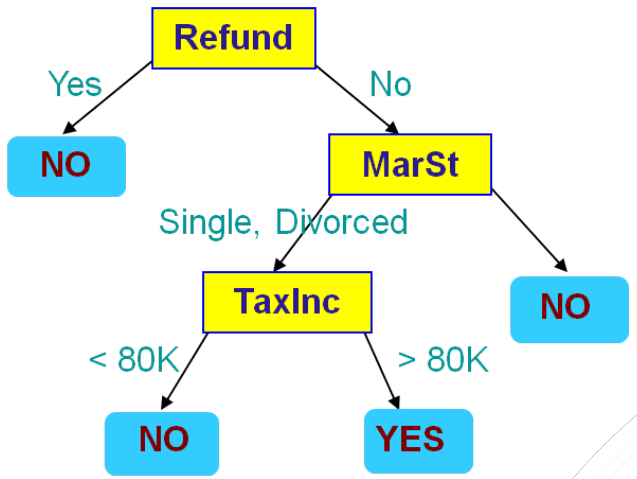
Ideas básicas

- *Los árboles de decisión tratan de encontrar una estructura jerárquica para explicar cómo diferentes partes del espacio de entrada corresponden con diferentes valores del atributo objeto.*
- *El arbol tiene tres tipos de nodos:*
  - *Nodo raiz por donde se empieza*
  - *Nodos internos cada uno de los cuales tiene un eje de entrada y dos o mas de salida que particiona el subespacio correspondiente a este nodo*
  - *Nodo hoja que no tiene ejes de salida y que está etiquetado con un valos del atributo objetivo*
- *En cada nodo que no sea hoja se divide parte del espacio de entrada en varios subconjuntos según el valor de un determinado atributo, hasta llegar a nodos hojas.*
- *En principio no se conoce la importancia de cada atributo en el proceso de clasificación por lo que es posible que se obtengan distintos resultados para un mismo problema*

Clasificación mediante Arboles de Decisión

Ejemplo: arbol a partir de datos

	categórico	categórico	continuo	clase
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

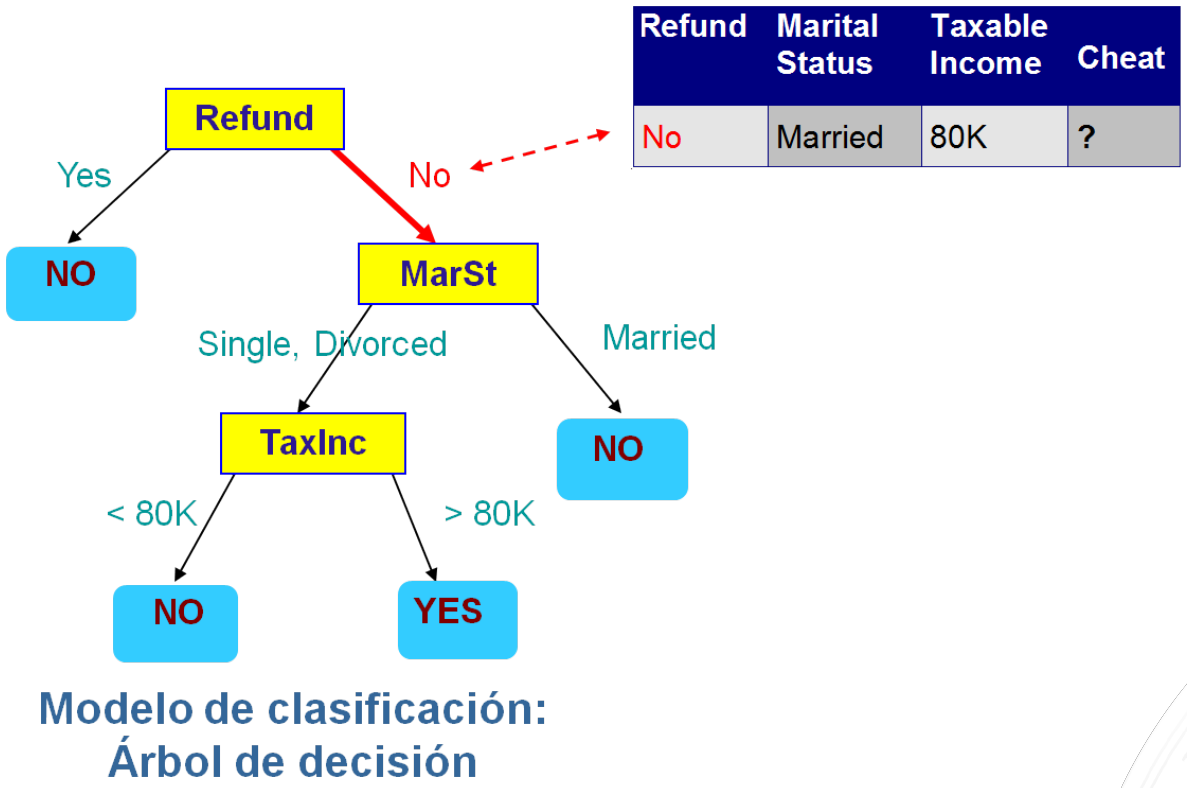


Conjunto de entrenamiento → **Modelo de clasificación: Árbol de decisión**

15

Clasificación mediante Arboles de Decisión

Ejemplo: clasificación de un item

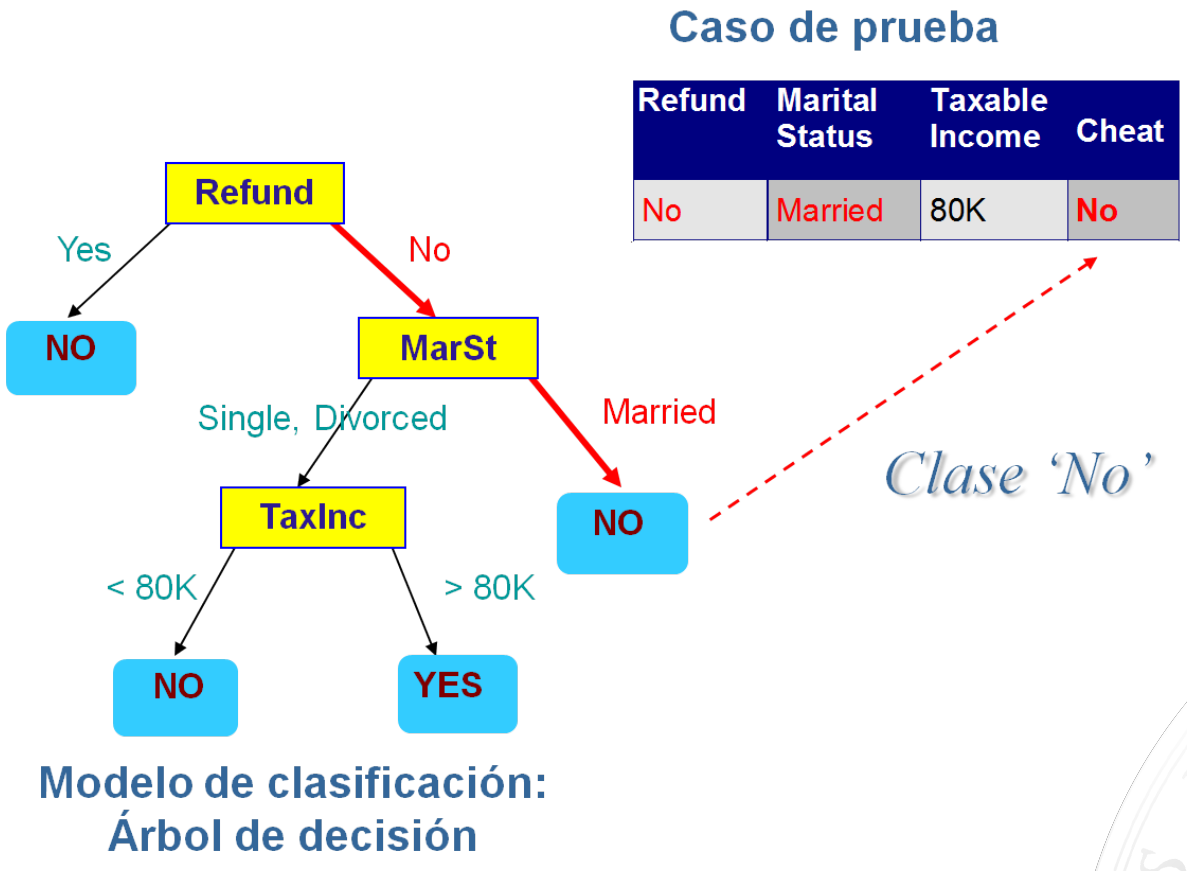


Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Modelo de clasificación: Árbol de decisión

Clasificación mediante Arboles de Decisión

Ejemplo: clasificación de un item

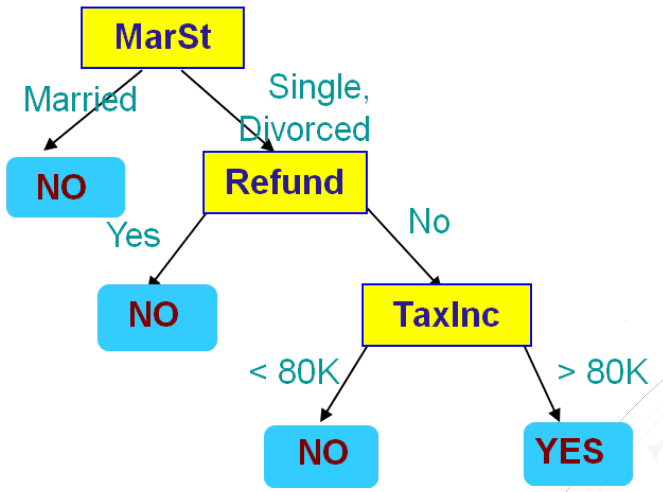


Clasificación mediante Arboles de Decisión

Ejemplo: otro arbol

	categórico	categórico	continuo	clase
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Podemos construir distintos árboles: ¿cuál es mejor?



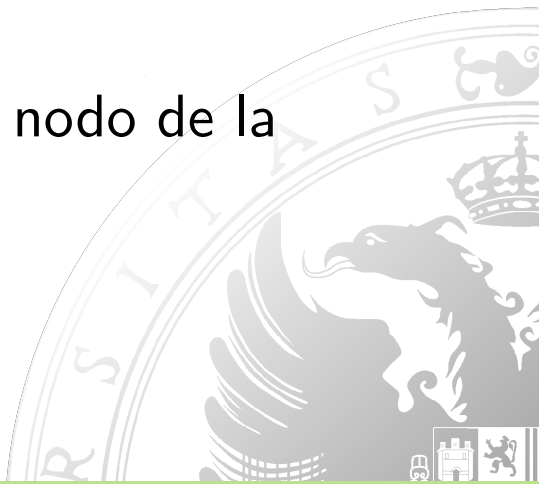
Conjunto de entrenamiento → **Modelo de clasificación: Árbol de decisión**

16

Construcción de Arboles de Decisión

Ideas básicas

- Encontrar un árbol de decisión óptimo no es trivial. Es 2^M donde M es el número de ejemplos
- Se busca un árbol razonablemente pequeño que explique adecuadamente el conjunto de entrenamiento
- Se utiliza una **Estrategia greedy** que convierte el problema en problema NP.
- Se parte de un nodo raiz y se va ramificando cada nodo de la "mejor " manera posible

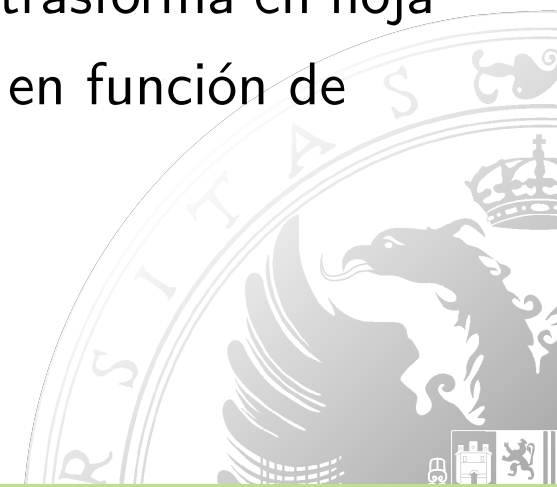


Construcción de Arboles de Decisión

Ideas básicas

⋆Algoritmo "divide y vencerás"

1. Comenzamos con todos los ejemplos de entrenamiento en la raíz del árbol de decisión.
2. Los ejemplos se van dividiendo en función del atributo que se seleccione para ramificar el árbol en cada nodo.
3. Si un nodo contiene ejemplos de sólo una clase se trasforma en hoja
4. Los atributos que se usan para ramificar se eligen en función de una heurística.
5. La forma de ramificar también



Construcción de Arboles de Decisión

Problemas en la aplicación de la heurística

¿Cuando se detiene la construccion de un arbol de decisión?

- Cuando todos los ejemplos que quedan pertenecen a la misma clase (se añade una hoja al árbol con la etiqueta de la clase).
- Cuando no quedan atributos por los que ramificar (se añade una hoja etiquetada con la clase más frecuente en el nodo).
- Cuando no nos quedan datos que clasificar.

Construcción de Arboles de Decisión

Problemas en la aplicación de la heurística

¿Dado un nodo que no es hoja, Como se particiona

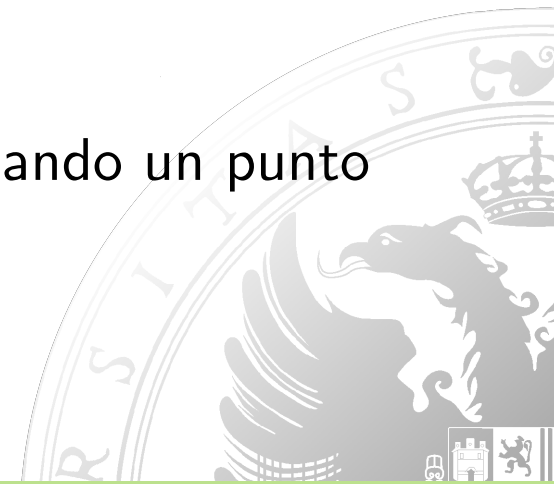
Nodos binarios No tiene problema

Nodos nominales Dos opciones:

- Particionar todos los valores (partición n-aria)
- Agrupar y transformar en binarios

Nodos ordinales Dos opciones:

- Particionar todos los valores
- Agrupar y transformar en binarios fijando un punto de corte ($\leq v, > v$)



Construcción de Arboles de Decisión

Problemas en la aplicación de la heurística

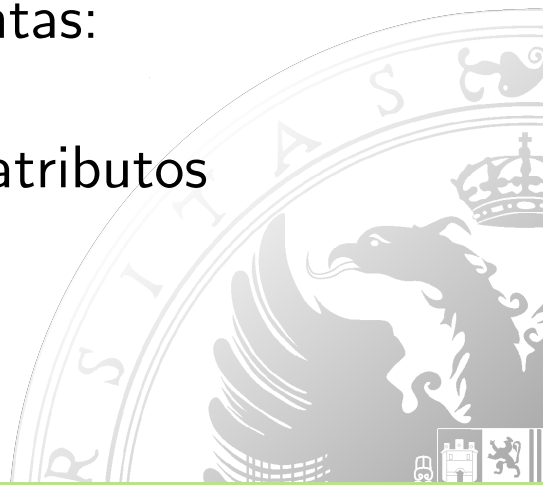
⋆**Criterio para la partición de nodos**

¿Dado un nodo que no es hoja, Como se particiona

Nodos numéricos Tenemos dos opciones:

- Discretizar el atributo y tratarlo como ordinal
- Agrupar y transformar en binarios fijando un punto de corte $(\leq v, > v)$

Distintos algoritmos utilzan formas de partición distintas:
CART solo binaria
ID3 sólo atributos discretos y particición n-aria para atributos categóricos
C4.5 partición n-aria y binaria para continuos, etc.

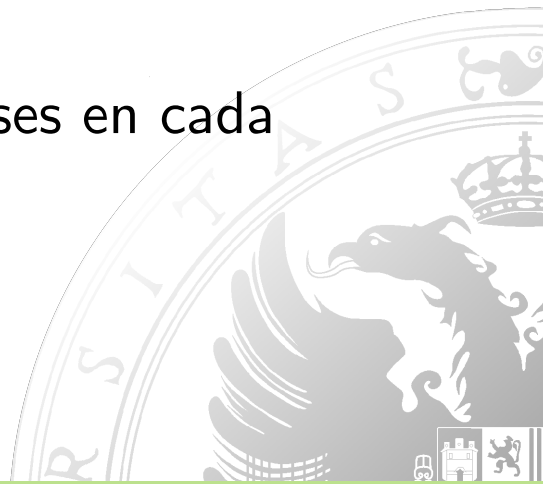


Construcción de Arboles de Decisión

Problemas en la aplicación de la heurística

Dado un nodo que no es hoja, ¿Qué atributo se elige para particionar?

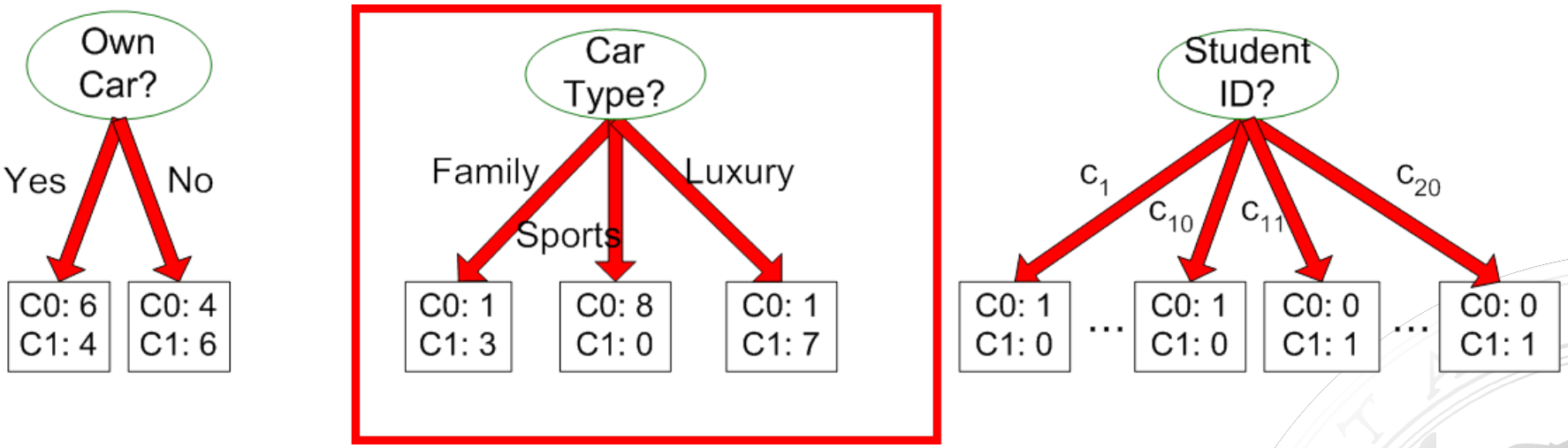
- Queremos un arbol pequeño. Se busca alcanzar nodos hoja cuanto antes.
- Necesitamos particiones con elementos de sólo una clase
- Se buscan particiones con nodos muy homogéneos
- Usaremos medidas basadas en la diversidad de clases en cada elemento de la partición. *medidas de impureza*



Construcción de Arboles de Decisión

Criterio de selección de atributos

Ejemplo

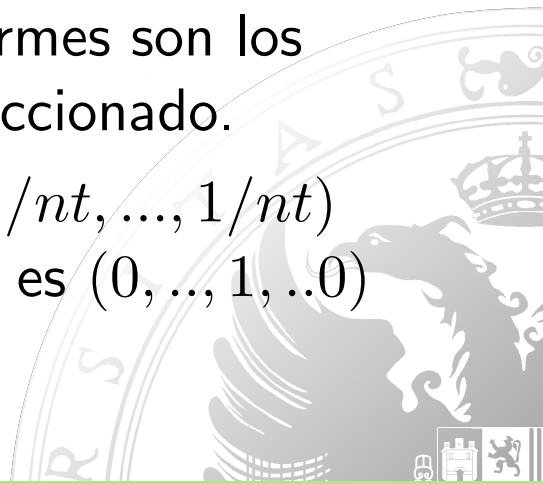


Construcción de Arboles de Decisión

Criterio de selección de atributos

⋆ Medidas de selección

- Sea $p(i/t)$ $i \in \{1, 2..c\}$ la fracción de items pertenecientes a la clase i que está en un determinado nodo t, obviamente c es el número de clases
- $p(i/t)$ es una aproximación de la probabilidad de encontrar un item de la clase i en la partición que t representa.
- De acuerdo con la idea anterior, cuanto mas uniformes son los valores de $p(i/t)$ menos deseable es t para ser seleccionado.
- El peor valor posible para $p(i/t)$ $i \in \{1, 2..c\}$ es $(1/nt, ..., 1/nt)$ donde nt es el número de elemento en t. El mejor es $(0, .., 1, ..0)$



Construcción de Arboles de Decisión

Criterio de selección de atributos

⋆ Medidas de selección basadas en entropía

Entropía $Entropia(t) = -\sum_{i=1}^{c} p(i/t) log_2(p(i/t))$

Indice de Gini $Gini = 1 - \sum_{i=1}^{c} p(i/t)^2$

Error de clasificación $error = 1 - max_i(p(i/t))$

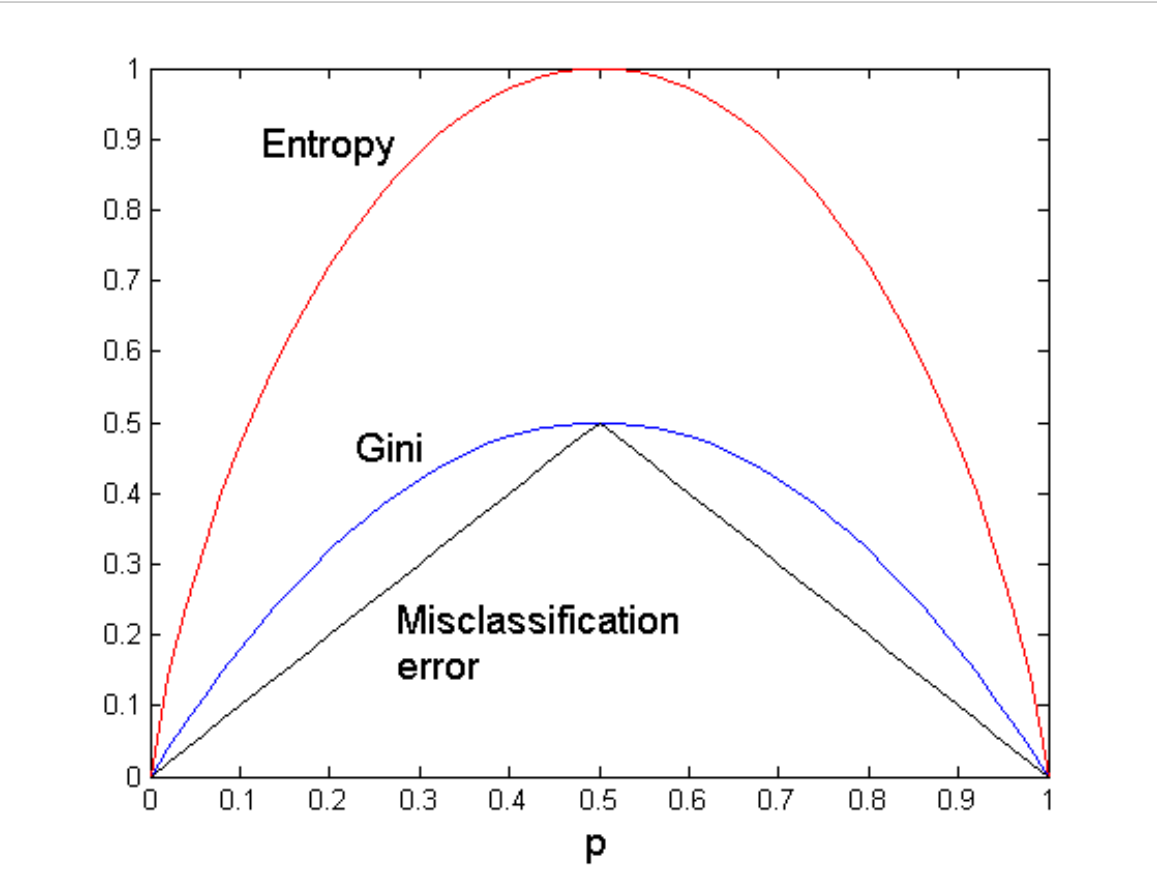
Ejemplo

Nodo	Distribución		Gini	Entropía	Error
N_1	Clase1	0	0	0	0
	Clase2	1			
N_2	Clase1	0.2	0.278	0.650	0.167
	Clase2	0.8			
N_1	Clase1	0.5	0.5	1	0.5
	Clase2	0.5			

Construcción de Arboles de Decisión

Criterio de selección de atributos

⋆ **Medidas de selección basadas en entropía, para dos clases**



Construcción de Arboles de Decisión

Criterio de selección de atributos

⋆ Ganancia de información

Para ver cómo funciona una división comparamos la medida de un nodo padre con la de los nodos hijos. Sean p un nodo padre $v_j, j = 1...k$ sus hijos, $N(p)$ número de elementos en el nodo p y $N(v_j)$ número de elementos en v_j. Definimos:

Ganancia $\triangle = I(p) - \sum_{j=1}^{k} \frac{N(v_j)}{N(p)} I(v_j)$, donde $I(.)$ es una de las medidas antes definidas. Sis $I(.)$ es la entropía se le denomina *Ganancia de información* $\triangle_{info}$

Proporción de ganancia de ganancia $Gainratio = \frac{\triangle_{info}}{SplitInfo}$ donde

$$SplitInfo = -\sum_{j=1}^{k} \frac{N(v_j)}{N(p)} log_2(\frac{N(v_j)}{N(p)}$$

$\triangle_{info}$ se usa en ID3 y $GainRatio$ en C4.5. CART, SLIQ..utilizan el índice de Gini

Construcción de Arboles de Decisión

Criterio de selección de atributos

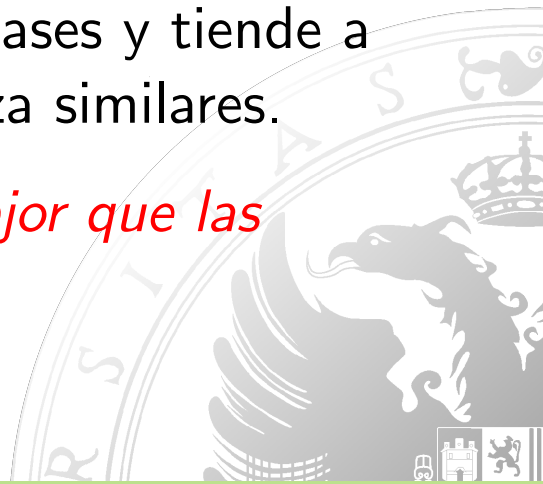
Comparación de reglas de división

Ganancia de información Sesgado hacia atributos con muchos valores diferentes.

Proporción de ganancia Tiende a preferir particiones poco equilibrada (con una partición mucho más grande que las otras)

Indice de Gini Funciona peor cuando hay muchas clases y tiende a favorecer particiones de tamaño y pureza similares.

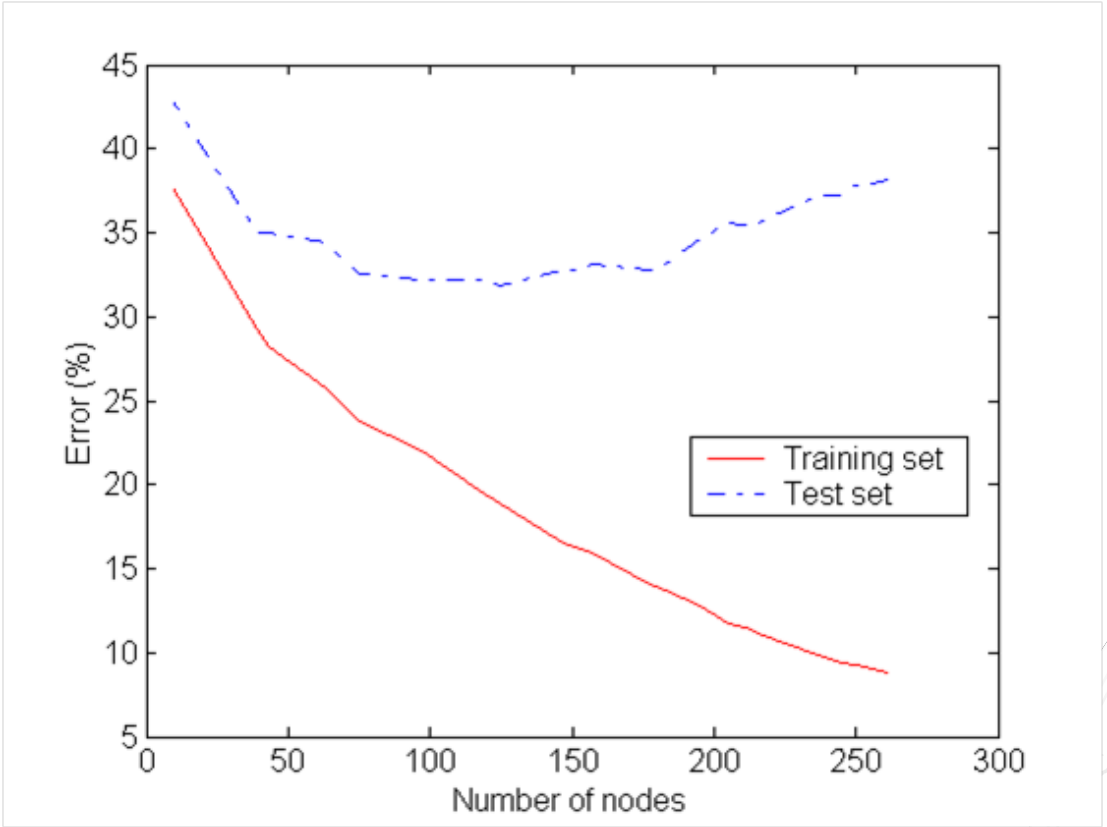
Ninguna regla de división es significativamente mejor que las demás



Construcción de Arboles de Decisión

Cuestiones adicionales: sobreaprendizaje

Cuanto mayor es su complejidad, los modelos de clasificación se ajustan más al conjunto de entrenamiento **sobreaprendizaje**.



Construcción de Arboles de Decisión

Cuestiones adicionales: sobreaprendizaje

Una solución al sobreaprendizaje son las *Técnicas de poda* que se se desarrollan para simplificar el árbol.
Para podar un árbol de decisión

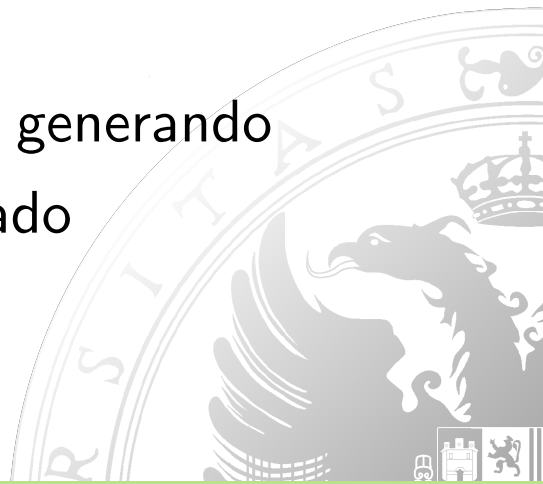
- Se sustituye un subárbol por un nodo hoja (correspondiente a la clase más frecuente en el subárbol)
- O bien, un subárbol por otro subárbol contenido en el primero.

Hay tecnicas de

Poda previa Se va reduciendo el árbol cuando se va generando

Poda a posteriori Se reduce el árbol una vez generado

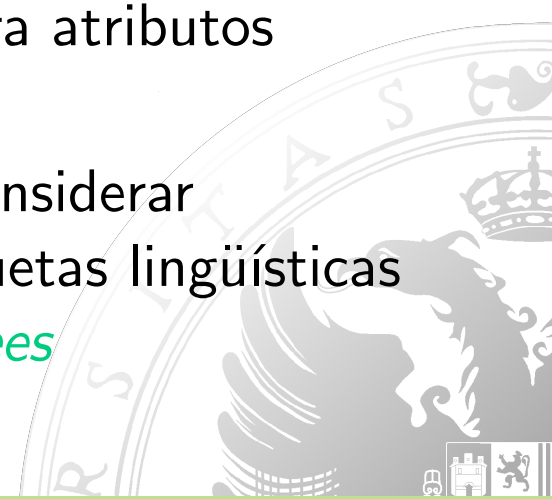
Para ver criterios ver bibliográfia básica



Construcción de Arboles de Decisión

Cuestiones adicionales

- La clasificación/predicción mediante árboles de decisión es una de las técnicas más estudiadas dentro de la clasificación.
- Existen numerosas variantes y ampliaciones de los algoritmos básicos en función de: mecanismos de partición del espacio, uso de técnicas de poda, uso de mecanismos adicionales como las reglas de asociación etc.
- Se ha extendido la idea para predecir métodos para atributos continuos *Arboles de regresión*
- Se han extendido los criterios de partición para considerar particiones difusas del dominio estableciendo etiquetas lingüísticas para los atributos discretizados *Fuzzy Decison Trees*



Construcción de Arboles de Decisión

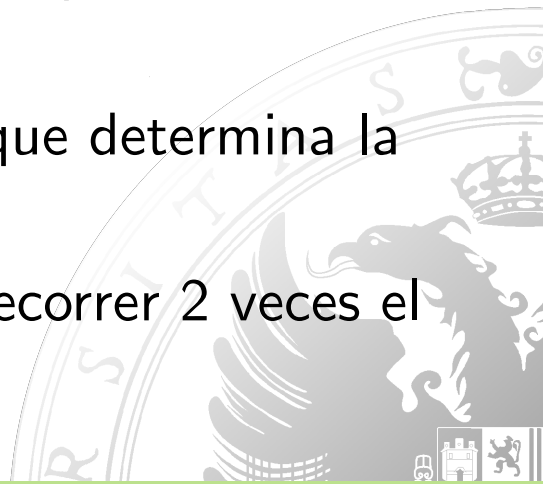
Cuestiones adicionales

Ventajas de los árboles de decisión

- Fácil interpretación (cuando son pequeños).
- Rapidez para clasificar nuevos datos.
- Precisión comparable a otras técnicas.

Algunos algoritmos eficientes y escalables

- PUBLIC (Rastogi and Shim, VLDB'1998) integra la poda en el proceso de construcción del árbol
- RainForest (Gehrke et al., VLDB'1998) separa lo que determina la escalabilidad del algoritmo
- BOAT (Gehrke et al., PODS'1999) sólo necesita recorrer 2 veces el conjunto de datos



Clasificación mediante reglas

Ideas básicas

Objetivo

Clasificar registros utilizando una colección de reglas "if then"

La forma de una regla es **Condicion** $\longrightarrow$ **y** Donde:

- **condición** es una conjunción de condiciones sobre el valor de varios atributos, también **antecedente**
- **y** es el valor de la clase también **consecuente**

Ejemplos de reglas

- Tipo sangre=caliente $\bigwedge$ Pone huevos= Si $\longrightarrow$ Pájaro
- Ingresos $\leqslant$ 30 $\bigwedge$ Devolución=si $\longrightarrow$ No evasor

Clasificación mediante reglas

Ideas básicas

Dada una regla r decimos que cubre una instancia x del dataset si dicha instancia satisface los antecedentes de la regla

Ejemplo

R1: (Viviparo = no) $\wedge$ (Puede volar = yes) $\rightarrow$ Pajaro

R2: (Viviparo = no) $\wedge$ (Acuatico = yes) $\rightarrow$ Pez

R3: ((Viviparo = yes) $\wedge$ (Sangre = caliente) $\rightarrow$ Mamífero

R4: ((Viviparo = no) $\wedge$ (Puede volar = no) $\rightarrow$ Reptil

R5: (Acuatico = a veces) $\rightarrow$ Anfibios

Nombre	Sangre	Vivíparo	Puede Volar	Acuatico	Clase
Halcón	Caliente	no	si	no	pajaro
Oso	Caliente	si	no	no	mamífero
Ornitorrinco	Caliente	no	no	A veces	mamífero
Lemur	Caliente	si	no	no	mamífero
Tortuga	fria	no	no	A veces	anfibio

Clasificación mediante reglas

Ideas básicas

Ejemplo

- Halcon es cubierto por R1
- Ornitorrinco y tortuga son cubiertos por R5

Una regla tiene:

Cobertura Proporción de registros que satisfacen sus antecedentes

Precisión Proporcion de registros que satisfacen antecedentes y consecuentes

Ejemplo

Regla	Cober.	Preci.
R1	1/5	1/5
R2	0	0
R3	2/5	2/5
R4	2/5	0
R5	2/5	1/5

Clasificación mediante reglas

Ideas básicas

Un conjunto test se clasifica registro a registro disparando las reglas que corresponden a los valores de cada uno de ellos, y anotando su clase

Ejemplo

Gorrión	Caliente	no	si	no	**pajaro**

El registro **dispara** *la regla R1*

Clasificación mediante reglas

Ideas básicas

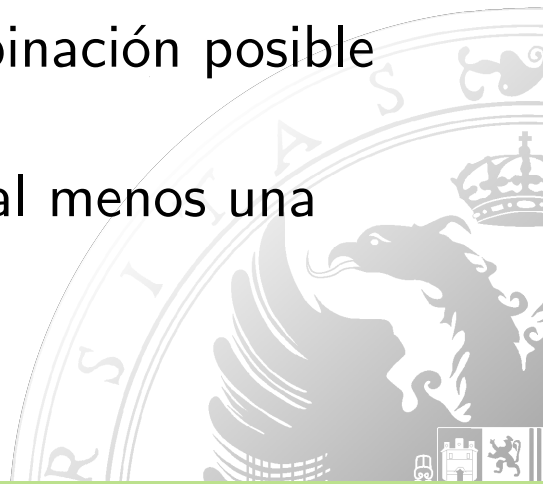
Con respecto a su aplicación un conjunto de reglas puede ser:

Mutuamente excluyentes Si:

- Cada regla puede aplicarse de forma independiente
- Cualquier registro está cubierto como mucho por una regla

Exhaustivo Si:

- Existe una regla para cualquier combinación posible de valores de atributos
- Cualquier registro está cubierto por al menos una regla



Clasificación mediante reglas

Ideas básicas

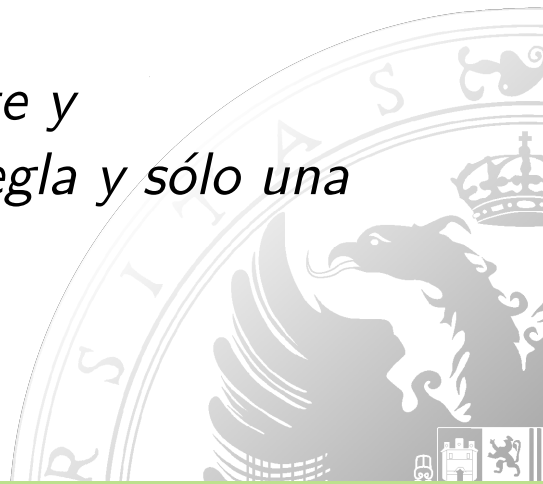
Ejemplo. conjunto de reglas mutuamente exclusivo y exhaustivo

r1: (Sangre= fria) $\rightarrow$ No mamífero

r2: (Sangre=caliente) $\wedge$ (Vivíparo = yes) $\rightarrow$ Mamífero

R3: ((Sangre = caliente) $\wedge$ (Viviparo = No) $\rightarrow$ No Mamífero

Si un conjunto de reglas es mutuamente excluyente y exhaustivo, cada registro a clasificar dispara una regla y sólo una



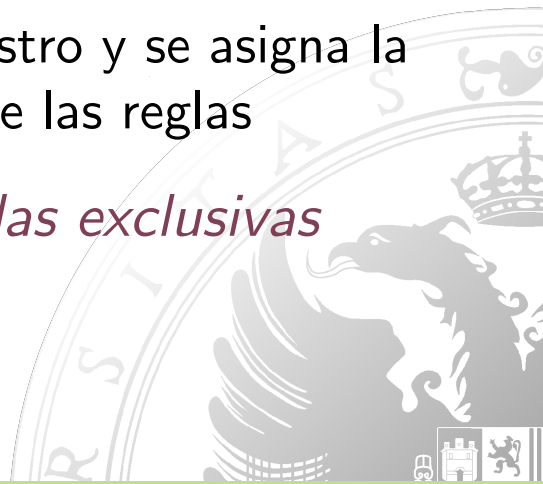
Clasificación mediante reglas

Ideas básicas

¿Qué hacer cuando un conjunto de reglas no tiene estas propiedades?

- Si no es exahustivo, se define una clase por defecto y se asignan a ella los registros no cubiertos
- Si no es exclusivo, un registro puede estar cubierto por varias reglas contradictorias. Soluciones:
  - Se ordenan las reglas por algún criterio: cobertura, precisión, clase que definen etc., y se aplica la regla más prioritaria
  - Se disparan todas las reglas correspondientes al registro y se asigna la clase "más votada", quizás ponderada por el peso de las reglas

La mayoría de los algoritmos que no producen reglas exclusivas siguen un criterio de ordenación



Clasificación mediante reglas

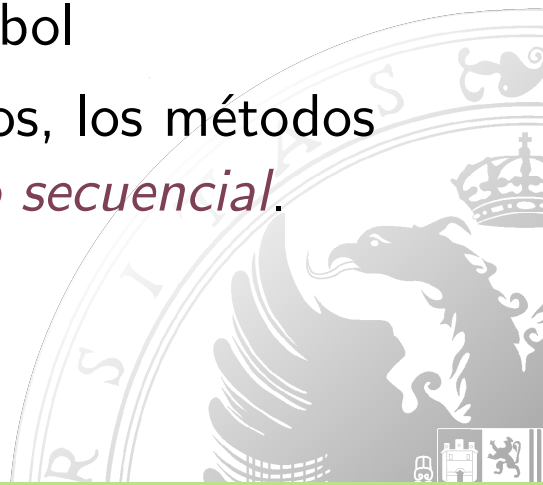
Extracción de reglas:ideas generales

Dado un conjunto de entrenamiento ¿Cómo extraer un conjunto de reglas?

A partir de un árbol de decisión Basta describir el árbol mediante un conjunto de reglas.

- Son mutuamente excluyentes
- Son exhaustivas
- Contienen toda la información del árbol

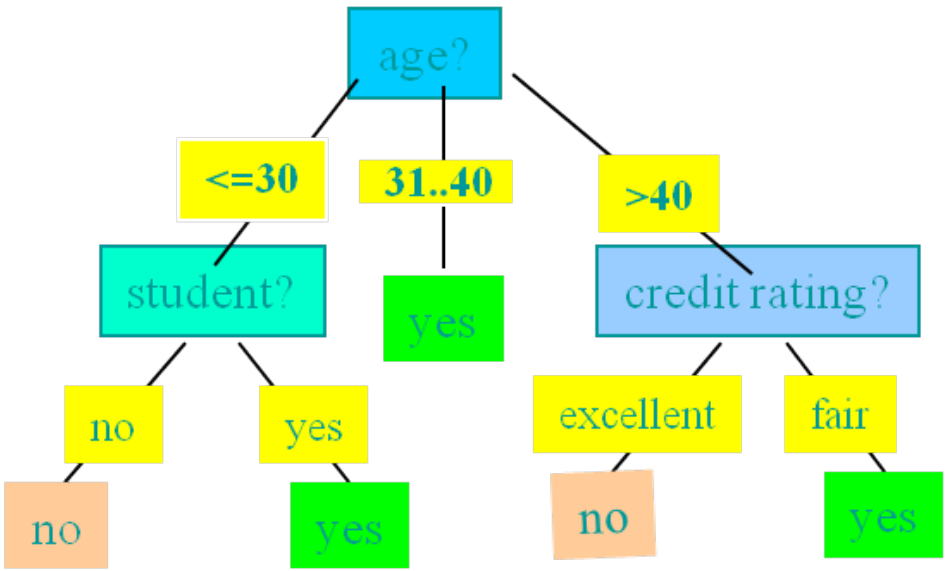
Métodos directos Actúan directamente sobre los datos, los métodos más conocidos son los de *recubrimiento secuencial*.
CN2, RIPPER y sus variantes etc.



Clasificación mediante reglas

Extracción de reglas mediante árboles de decisión

Ejemplo



IF (age<=30) AND (student=no) THEN buys_computer = no

IF (age<=30) AND (student=yes) THEN buys_computer = yes

IF (30<age<=40) THEN buys_computer = yes

IF (age>40) AND (credit_rating=excellent) THEN buys_computer = no

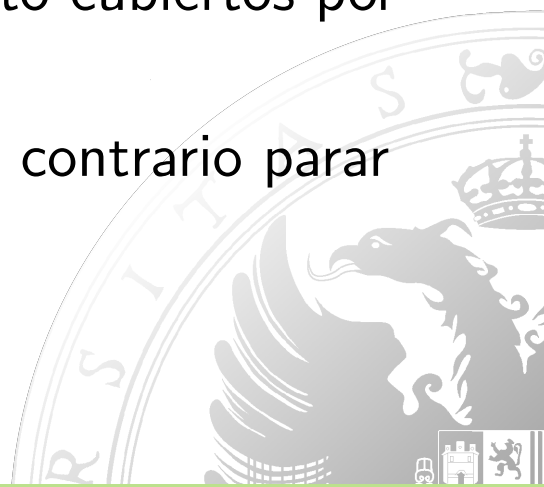
IF (age>40) AND (credit_rating=fair)THEN buys_computer = yes

Clasificación mediante reglas

Extracción directa reglas: ideas básicas

El proceso básico es el de **Recubrimiento secuencial**

1. Empezar con un conjunto de reglas vacío
2. Generar la **mejor regla** que cubre una clase concreta
3. Añadir la regla al conjunto aprendido
4. Eliminar los ejemplos del conjunto de entrenamiento cubiertos por la regla
5. Si no se cumple la **regla de parada** ir a 2 en caso contrario parar

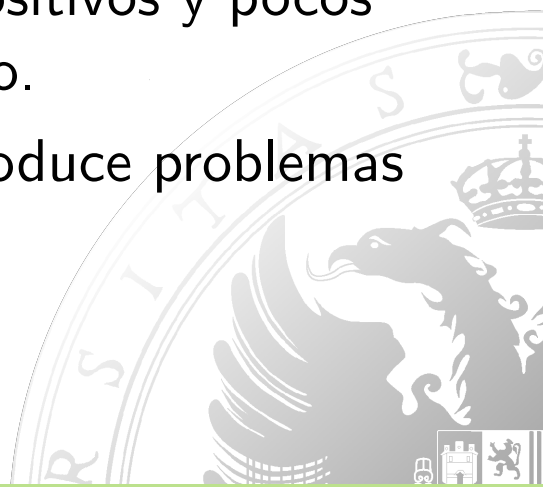


Clasificación mediante reglas

Extracción directa reglas: ideas básicas

Para extraer la mejor regla:

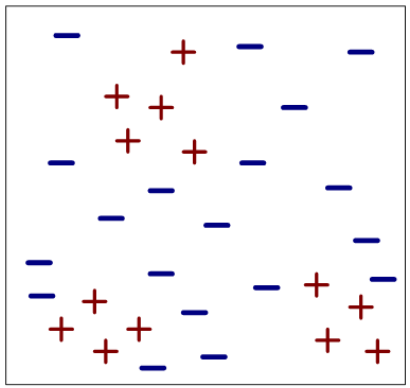
1. Se ordenan las clases (hay criterios de mayor a menor o al contrario)
2. Se consideran ejemplos positivos los de esa clase y negativos el resto
3. La mejor regla es a que cubre muchos ejemplos positivos y pocos negativos. Se usan medidas de evaluación para ello.
4. Es posible que una regla necesite ser podada si produce problemas de sobreaprendizaje.



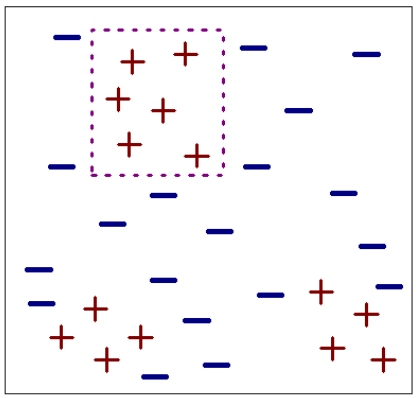
Clasificación mediante reglas

Extracción directa reglas: ideas básicas

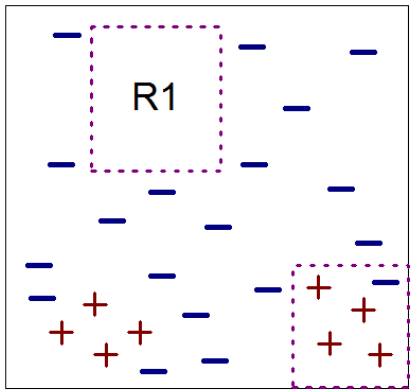
Ejemplo de selección de reglas



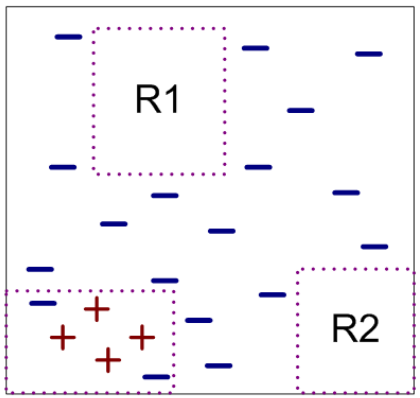
(i) Original Data



(ii) Step 1



(iii) Step 2



(iv) Step 3

Clasificación mediante reglas

Extracción directa reglas: ideas básicas

Algunos algoritmos

- FOIL (Quinlan, Machine Learning, 1990)
- CN2 (Clark and Boswell, EWSL'1991)
- RIPPER (Cohen, ICML'1995)
- PNrule (Joshi, Agarwal and Kumar, SIGMOD'2001

Clasificación Bayesiana

Ideas básicas

- Existen problemas en los que la relación entre items y clases tiene una componente aleatoria
- Incluso items (instancias) con valores iguales en los atributos pueden pertenecer a clases distintas (diagnóstico de enfermedades según síntomas)
- **Principio básico** Si no se puede asegurar a qué clase pertenece una instancia,**asignarle la clase a la que tiene mayor probabilidad de pertenecer**

Clasificación Bayesiana

Ideas básicas

- Dos problemas:
  - **A** Dada una instancia determinada, ¿Cómo se estima la clase más probable a la que pertenece? *Mediante el teorema de Bayes*
  - **B** ¿Como se almacenan/calculan las clases más probables según las posibles combinaciones de valores de los atributos de forma eficiente? *Hipótesis simplificadoras*

    Hipótesis de independencia para atributos discretos → Conduce a los métodos "Naïve Bayes"

    Hipótesis de distribución normal conjunta para atributos continuos → Conduce al Análisis Discriminante y sus variantes

Clasificación Bayesiana

Modelo probabilístico: teorema de Bayes

- Dados sucesos aleatorio A y C tenemos que:

$$P(C|A) = \frac{P(A,C)}{P(C)} \; P(A|C) = \frac{P(A,C)}{P(A)} \implies$$

$$\implies P(C|A) = \frac{P(A|C)P(C)}{P(A)}$$

- *Ejemplo de uso*
  - Se sabe que la meningitis causa rigidez de cuello en el 50% de los casos.
  - Se sabe que la probabilidad de tener meningitis es de $1/50000$ y la de que un paciente tenga rigidez de cuello de $1/20$.
  - Entonces:

$$P(Men|Rig) = \frac{P(Rig|Men)P(Men)}{P(Rig)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Clasificación Bayesiana

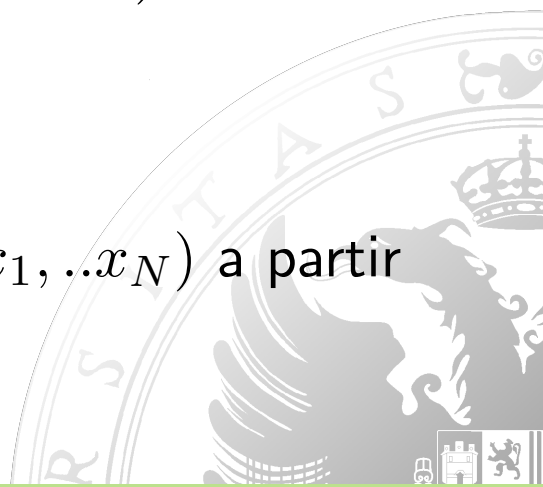
Modelo probabilístico

- Supongamos que tanto los atributos $X_1..X_N$ como la clase Y son variables aleatorias.
- Dada una instancia(item) con valores de atributos $x_1..x_N$ queremos predecir el valor de su clase y
- Específicamente queremos encontrar el valor de y que maximiza la expresión:

$$Prob(Y = y|X_1 = x_1, X_2 = x_2, ..., X_n = x_N)$$

para simplificar $P(y|x_1, x_2.., x_N)$

- *Problema*:
  ¿Podemos calcular $P(y|x_1, x_2.., x_N)$, conocidos $(x_1, ..x_N)$ a partir de los datos?
- *Solución* El uso del teorema de Bayes.



Clasificación Bayesiana

Modelo probabilístico: algoritmo básico

1. Para todo $y \in dom(Y)$ calcular:

$$P(y|x_1, x_2.., x_n) = \frac{P(x_1, x_2.., x_N|y)P(y)}{P(x_1..x_N)}$$

2. Elegir $\hat{y}$ tal que

$$P(\hat{y}|x_1, x_2.., x_n) = max_{y \in dom(Y)} \frac{P(x_1, x_2.., x_N|y)P(y)}{P(x_1..x_N)}$$

3. Lo realmente es equivalente a elegir $\hat{y}$ tal que

$$P(\hat{y}|x_1, x_2.., x_n) = max_{y \in dom(Y)} P(x_1, x_2.., x_N|y)P(y)$$

4. *Problema*. ¿Cómo estimar $P(x_1, x_2.., x_N|y)$?. En principio se trata de distribuciones conjuntas de N variables aleatorias, condicionadas a cada valor del dominio de clases

Clasificación Bayesiana

Clasificador Naïve Bayes

Se asume la independencia de condicionada entre los atributos $X_1..X_N$ de forma que:

$$\forall y \in Dom(Y)\, P(x_1, x_2.., x_N|y) = P(x_1|y)P(x_2|y)..P(x_N|y)$$

1. $\forall y \in Dom(Y)\, \forall j \in \{1, ..N\}$ se puede estimar $P(x_j|y)$ y $P(y)$ utilizando el conjunto de entrenamiento
2. Dado un nuevo item de valores $(v_1..v_N)$ se clasifica en la clase z tal que:

$$P(z)P(v_1|z)..P(v_N|z) = max_{y \in dom(Y)} P(y)P(v_1|y)..P(v_N|y)$$

Clasificación Bayesiana

Clasificador Naïve Bayes: ejemplo sencillo

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Probabilidad de clases

$P(y) = M_y/M$, P(Si)= 3/10, P(No)=7/10

Probabilidad de atributos discretos

$$P(x_j|y) = m_{x_j y}/m_y$$

P(Status=married|no)=4/7

Probabilidad de atributos continuos

Hipótesis de normalidad

$$P(x_j|y) = (1/\sqrt{2\pi\sigma_{jy}^2}) \exp \frac{(x_j-\mu_{jy})^2}{2\sigma_{jy}^2}$$

P(Income=120|No)= 0.0072

Clasificación Bayesiana

Clasificador Naïve Bayes: ejemplo sencillo

Sea X=(Refund=NO,Married,Income=120)

naive Bayes Classifier:

P(Refund=Yes|No) = 3/7
P(Refund=No|No) = 4/7
P(Refund=Yes|Yes) = 0
P(Refund=No|Yes) = 1
P(Marital Status=Single|No) = 2/7
P(Marital Status=Divorced|No)=1/7
P(Marital Status=Married|No) = 4/7
P(Marital Status=Single|Yes) = 2/7
P(Marital Status=Divorced|Yes)=1/7
P(Marital Status=Married|Yes) = 0

For taxable income:
If class=No: sample mean=110
sample variance=2975
If class=Yes: sample mean=90
sample variance=25

P(X|Class=No) = P(Refund=No|Class=No)
× P(Married| Class=No)
× P(Income=120K| Class=No)
= 4/7 × 4/7 × 0.0072 = 0.0024

P(X|Class=Yes) = P(Refund=No| Class=Yes)
× P(Married| Class=Yes)
× P(Income=120K| Class=Yes)
= 1 × 0 × 1.2 × 10^{-9} = 0

Since P(X|No)P(No) > P(X|Yes)P(Yes)

Therefore P(No|X) > P(Yes|X)
=> Class = No

Clasificación Bayesiana

Clasificador Naïve Bayes: estimación de las probabilidades de atributos discretos

En general $\forall y \in Dom(Y)\, \forall j \in \{1, ..N\} P(x_j|y)$ se estima:

$$P(x_j|y) = \frac{\gamma + m_{x_j y}}{\gamma m_{X_j} + m_y}$$

donde m_{X_j} es el número de elementos en $dom(X_j)$ y $p(y)$ se estima como:

$$p(y) = \frac{\gamma + m_y}{\gamma m_Y + m}$$

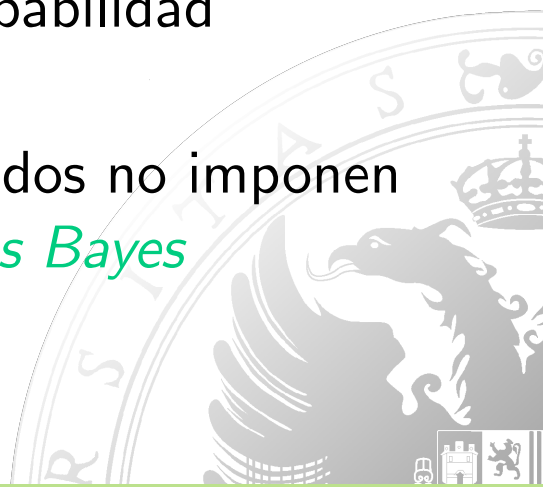
donde m_Y es el número de elementos que tiene $dom(Y)$ y m el numero de elementos del dataset

- La constante γ se denomina *corrección de Laplace*.
- Habitualmente se toma igual a cero; pero para tratar casos de valores de atributos no existentes se toma igual a 1 o 1/2
- Se usa cuando el conjunto de entrenamiento es pequeño

Clasificación Bayesiana

Clasificador Naïve Bayes: estimación de las probabilidades de atributos continuos

- Un atributos continuo X_j se considera distribuido $N(\mu_{jY}, \sigma^2_{jy})$ todo valor de clase y. Y su probabilidad condicionada viene dada por $f(x_j|y) = N(\mu_{jY}, \sigma^2_{jy})(x_j)$
- Cuando se tiene un conjunto de atributos continuos $\bar{X}$, se puede evitar la hipótesis de independencia condicionada suponiendo que, $\forall y$, $(\bar{X}|y)$ se distribuye según una normal multivariante $N(\mu_{\bar{X}|y}, \sum_{\bar{X}|y})$, se puede entonces calcular la probabilidad condicionada conjunta $f(\bar{x}|y)$.
- Cuando sólo se tiene atributos numéricos los métodos no imponen *hipótesis de independencia* y tenemos *clasificadores Bayes completos*

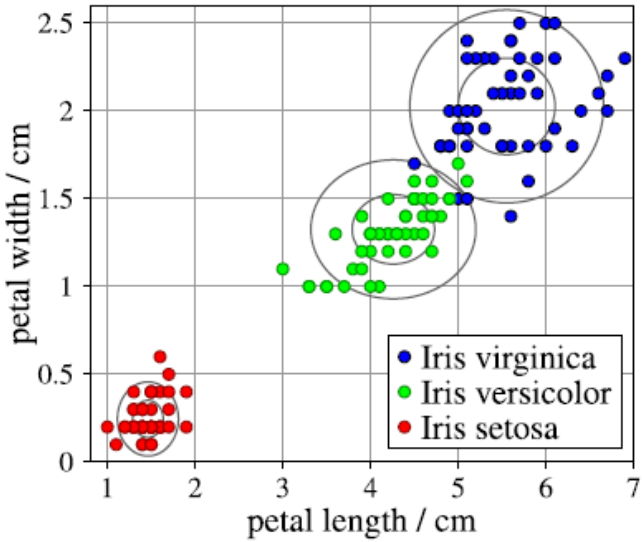


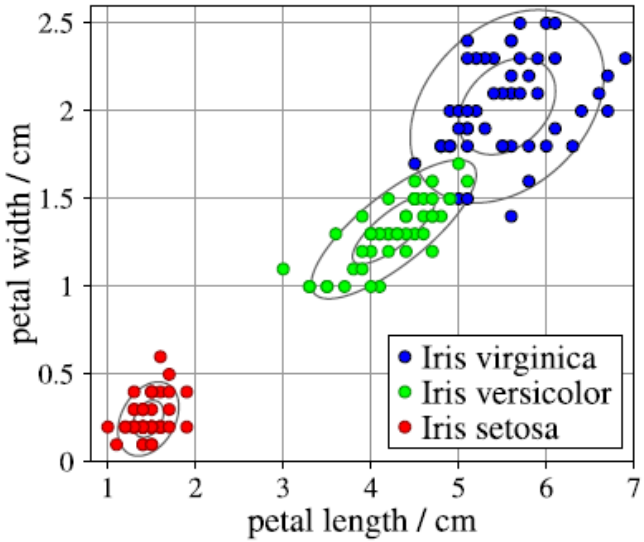
Clasificación Bayesiana

Clasificador Naïve Bayes: estimación de las probabilidades de atributos continuos

Ejemplo

Iris type	Iris setosa	Iris versicolor	Iris virginica
Prior probability	0.333	0.333	0.333
Petal length	1.46 ± 0.17	4.26 ± 0.46	5.55 ± 0.55
Petal width	0.24 ± 0.11	1.33 ± 0.20	2.03 ± 0.27

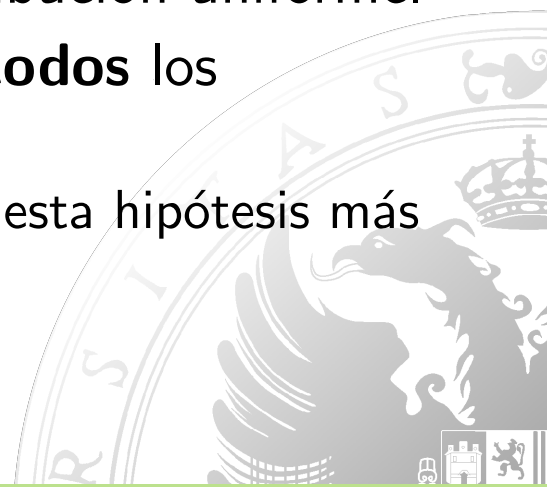




Clasificación Bayesiana

Clasificador Naïve Bayes: resumen

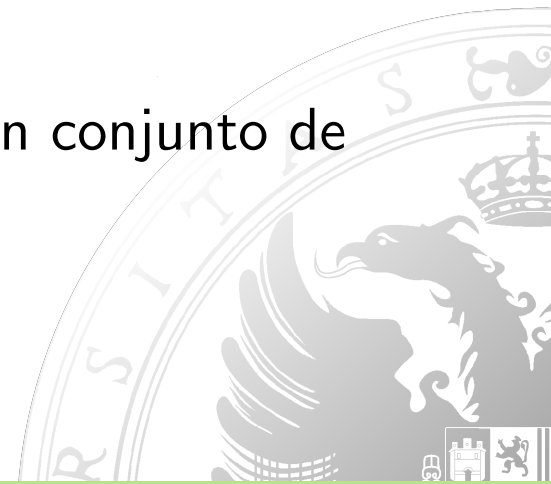
- Son robustos frente a puntos ruido aislados ya que trabajan en frecuencias y medias
- Permiten manejar valores pedidos ignorando las instancias que los tengan en la fase de estimación de probabilidades.
- Son robustos frente a atributos irrelevantes ya que si un atributo X no tiene influencia en Y $P(X|Y)$ tiende a la distribución uniforme.
- La hipótesis de independencia condicionada para **todos** los atributos puede ser muy fuerte.
  - Las redes bayesianas generalizan el modelo y hacen esta hipótesis más flexible



Clasificación Bayesiana

El análisis discriminante

- Es un caso particular de *clasificador Bayes completo*
- Hipótesis Simplificadoras:
  - Atributos Numéricos
  - Distribución normal mutivariante. Con restricciones:
    - Matriz de covarianza igual por clases
    - Medias muy diferentes
  - Inicialmente sólo dos clases
- El calculo de la clase óptima se basa en calcular un conjunto de hiperplanos que dividan el espacio por clases.



Clasificación basadas en instancias

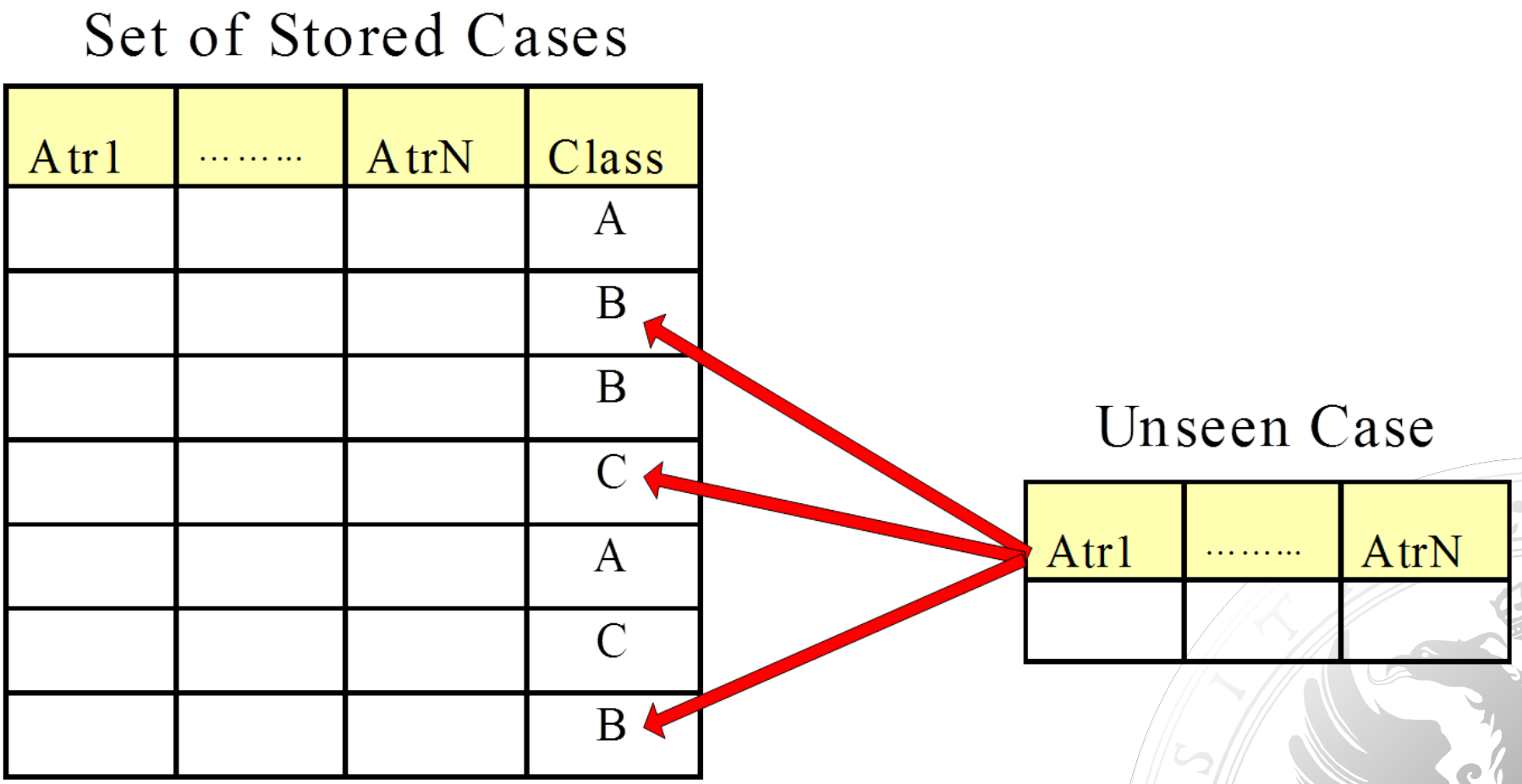
- Los clasificadores estudiados hasta ahora, trabajan en dos etapas:
  - Inductiva Aprenden el modelo de clasificación
  - Deductiva Aplican el modelo a los ejemplos del conjunto test
- Son "'clasificadores ansiosos", (**eager learners)**

Idea básica

¿Por qué no almacenar todo el conjunto de entrenamiento y cuando llegue un ejemplo test buscar, los items que "más se le parecen" y asignarle su clase?.

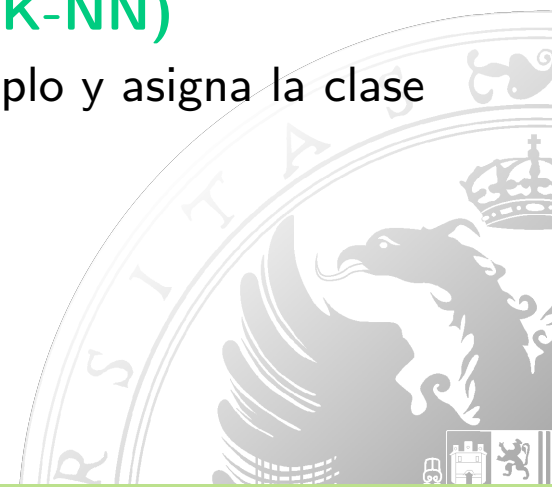
Son métodos "perezosos" **Lazy Learners**

Clasificación basadas en instancias



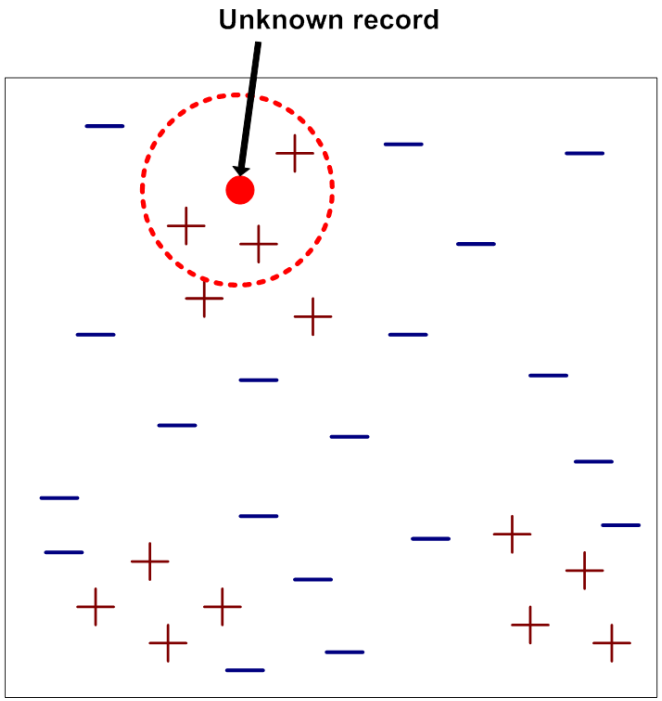
Clasificación basadas en instancias

- Obviamente son métodos predictivos. No explican nada.
- Ejemplos:
  - Los "clasificadores memorísticos" (**Rote Learners**)
    - Almacena el conjunto de entrenamiento y solo asigna una clase a un ejemplo cuando hay un item de entrenamiento que es exactamente igual a él.
  - **K-vecinos más cercanos (K nearest neighbor) (K-NN)**
    - Selecciona los k-items que "más se parecen" al ejemplo y asigna la clase más frecuente o "más importante"



K-vecinos más cercanos (K-NN)

Ideas básicas



Requerimientos

Distancia entre registros $d(.,.)$

El valor k fijado

Algoritmo

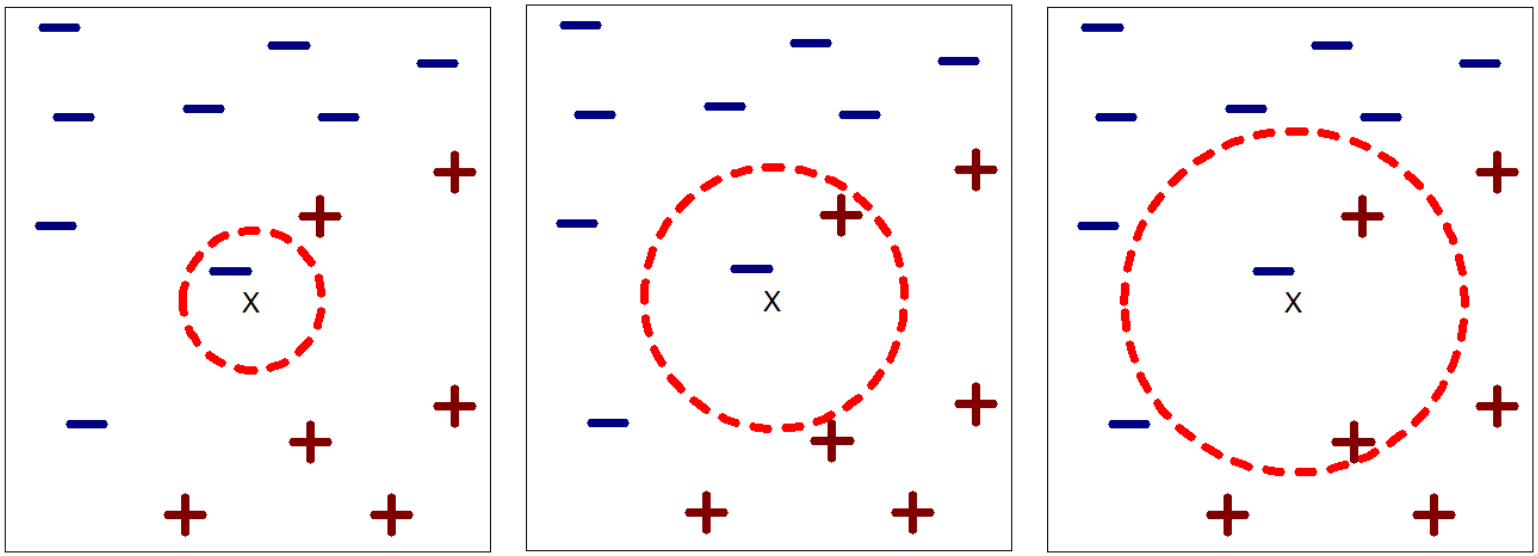
1.- Sea x calcular $d(x,e)$, $\forall e \in E$

2.- Identificar $\{e_i, i = 1...k\}$
k vecinos más cercanos a x

3.- Con las clases de $\{e_i, i = 1...k\}$,
obtener la clase de x
Mediante mayoría o mayoría ponderada

K-vecinos más cercanos (K-NN)

Ideas básicas



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

Es importante fijar el valor de k. Ya que puede conducir a sobreaprendizaje o a error.

K-vecinos más cercanos (K-NN)

Problemas de aplicación

La función de distancia .

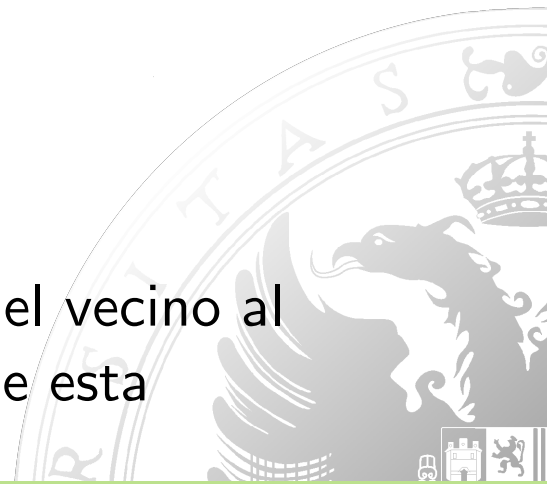
- El método está pensado para atributos numéricos
- Se pueden usar las distancias propuestas en el tema de agrupamiento
- Habra que tener en cuenta problemas de escala

El valor de k .

- Lo mejor es obtenerlo mediante **validación cruzada** (*se verá posteriormente*)

Mecanismo de selección de clases .

- Se suele elegir la clase mayoritaria
- Se puede ponderar por la distancia del vecino al punto o funciones mas sofisticadas de esta

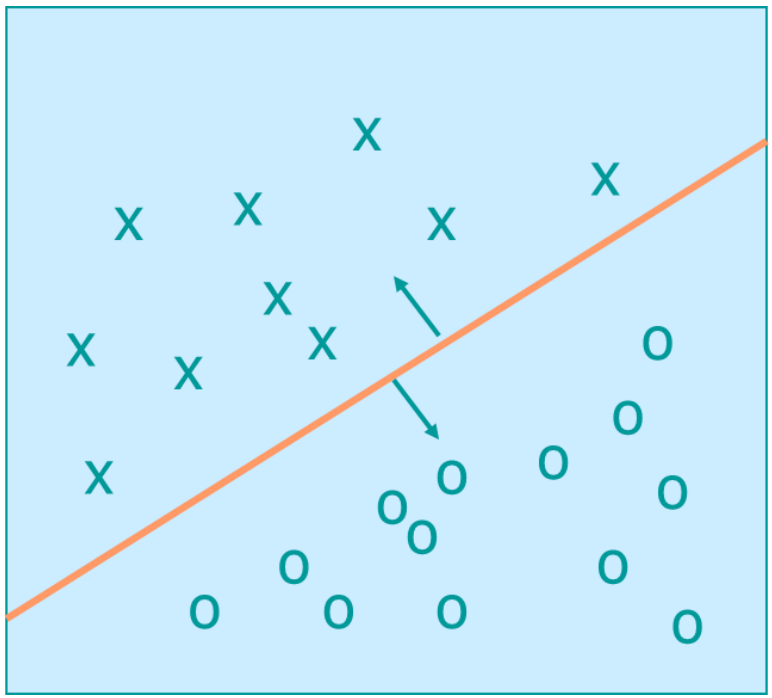


Otros modelos predictivos de clasificación

Clasificadores basados en Redes Neuronales

Se han estudiado en otras asignaturas

SVMs Support Vector Machines



Otros modelos predictivos de clasificación

SVMs Support Vector Machines

Ventajas .

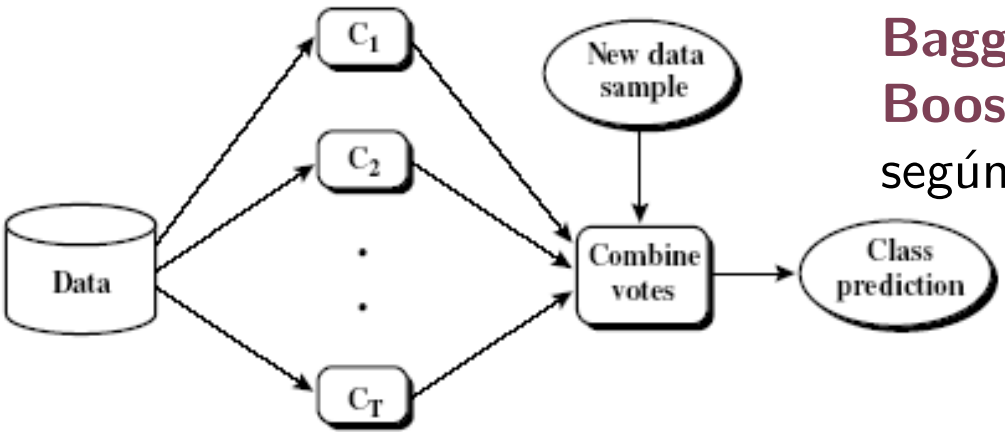
- Precisión alta
- Robustez frente al ruido

Inconvenientes

- Costosos de entrenar. (Poco escalables y eficientes)
- Dificiles de interpretar (Aumentan la dimension del espacio para separar las clases por hierplanos)

Otros modelos de clasificación

"Emsembles"



Combinan varios modelos
para mejorar la precisión del clasificador
Para decidir qué clase se asigna:
Bagging Votación por mayoria
Boosting Votación ponderada
según calidad del clasificador (*AdaBoost*)

Evaluación de modelos de clasificación

Ideas básicas

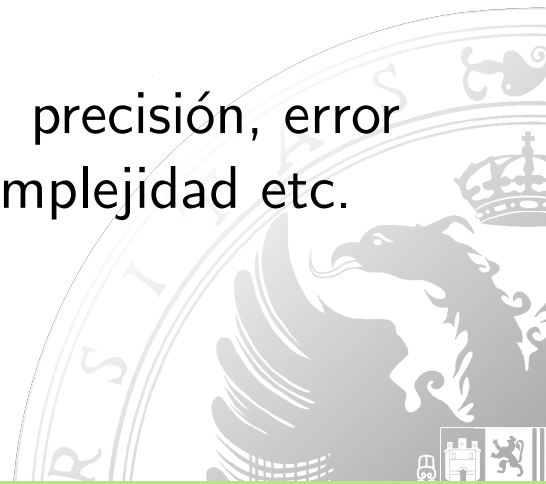
Pasos del proceso de clasificación:

1.- Se considera un data set con valores en Y.**Conjunto de entrenamiento**

2.- Se construye (aprende) el modelos de clasificación

3.- Se prueba en otro dataset distinto **Conjunto test** caculando los valores de Y^{pred}

4.- **Se evalúa en el modelo** según distintos criterios: precisión, error de clasificación, escalabilidad, interpretabilidad, complejidad etc.



Evaluación de modelos de clasificación

Ideas básicas

—*Distintos aspectos en la evaluación de modelos*

Métricas Son medidas de la "calidad" de un proceso de clasificación.

Modelos Sirven para estimar de forma fiable las medidas de calidad.

Comparación Son técnicas que permiten comparar el rendimiento relativo de dos modelos de clasificación

Evaluacion de modelos de clasificación

Ideas básicas

—*Criterios de para las medidas de calidad*

- **Precisión** ¿Cómo de bien clasifica el modelo?
- **Eficiencia** Tiempo necesario para construir/usar el clasificador
- **Robustez** Frente a ruido y valores nulos
- **Escalabilidad** ¿Admite grandes datasets?
- **Interpretabilidad** ¿Explica el modelo?
- **Complejidad** Arbol con muchos nodos etc.

Los dos primeros pueden ser métricas mientras que los últimos sólo se pueden medir en algunos casos.

Las medidas de precisión serán clave

Evaluación de modelos de clasificación

Medidas de Precisión

Sean: M un clasificador, y $x_i = x_{i1}, ..x_{iN}$ un ejemplo de las variable independientes de un conjunto test $T = \{x_1y_1, ..x_ny_n\}$.
Sea $\hat{y}_i = M(x_i)$ el resultado de aplicar el proceso de M a x_i .
Definimos:

Precisión de M .

$$Acc = 1/n \sum_{i=1}^{n} I(y_i = \hat{y}_i)$$

donde $I(e)$ es igual a 1 si e es cierta e igual a 0 si e es falsa.

Tasa de error de M .

$$Er = 1/n \sum_{i=1}^{n} I(y_i \neq \hat{y}_i) = 1 - Acc$$

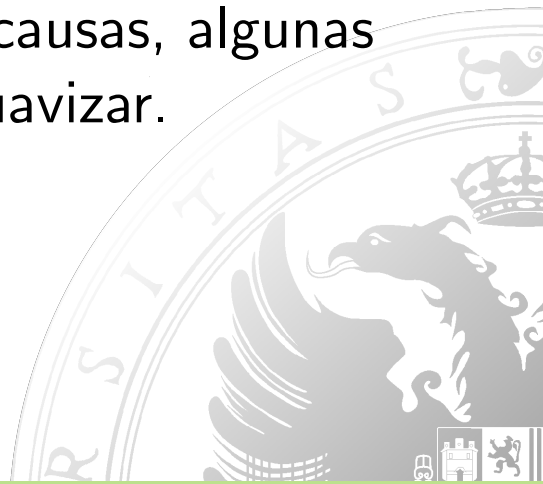
Evaluación de modelos de clasificación

Medidas de Precisión

Parece que lo mejor es tener una baja tasa de error/alta precisión pero:

> *Si se trata de ajustar excesivamente el modelo al conjunto de entrenamiento sólo aprende este.* **Sobreaprendizaje**

La aparición del sobreaprendizaje de debe a diversas causas, algunas dependientes de los modelos; pero otras se pueden suavizar.



Evaluacion de modelos de clasificación

Medidas de Precisión

Algunas consideraciones sobre el sobreaprendizaje

- Cuanto mayor sea la complejidad de un modelo de clasificación, más se ajusta excesivamente al conjunto de entrenamiento (en árboles de decisión)
- Otra causa de sobreaprendizaje es la presencia de puntos ruido (en modelos que particionan el espacio: análisis discriminante, SVM p.e.)
- Por último a escasez de puntos en una clase frente a otras puede dar problemas de sobreaprendizaje, ya que la precisión no tiene en cuenta el tamaño de las clases

Para evitar este último problema hay que tener en cuenta el peso de las clases en las medidas de precisión, para ello
Matrices de Confusión

Evaluacion de modelos de clasificación

Medidas de Precisión: matrices de confusión

Dado un proceso de clasificación si definimos n_{ij} número de elementos que se asignan a la clase i cuando están en la clase j tenemos:
Matriz de confusión

Predichos \ ciertos	y_1	y_2	$\cdots\cdots$	y_k	tot_{pred}
$\hat{y}_1$	n_{11}	n_{12}	$\cdots\cdots$	n_{1k}	m_1
$\vdots$	$\vdots$	$\vdots$	$\cdots\cdots$	$\vdots$	$\vdots$
$\hat{y}_k$	n_{k1}	n_{k2}	$\cdots\cdots$	n_{kk}	m_k
tot_{cier}	n_1	n_2	$\cdots\cdots$	n_k	n

Las matrices de confusión permiten definir medidas asociadas a las clases y medidas globales ponderadas

Evaluacion de modelos de clasificación

Medidas de Precisión: matrices de confusión

Se pueden definir:

Precisión de una clase $\forall i \in \{1..k\}\ , acc_i = n_{ii}/m_i$

"Recall" de una clase $\forall i \in \{1..k\}\ , recc_i = n_{ii}/n_i$

F medida de una clase $\forall i \in \{1..k\}\ , F_i = 2n_{ii}/(m_i + n_i)$

La precisión y recall global siguen siendo las mismas pero se puede definir:

F global $F = 1/k \sum_{i=1}^{k} F_i$

Evaluacion de modelos de clasificación

Medidas de Precisión: matrices de confusión

Ejemplo

Datos del iris utilizando "sepal length" y "sepal width".

Clasificador Naïve Bayes. 120 ejemplos entrenamiento y 30 datos test.

Datos globales $acc = 0.733$ $er = 0.267$

Matriz de confusion

Predichos \ ciertos	Setosa	Versicolor	Viginica	
Setosa	10	0	0	10
Versicolor	0	7	5	12
Virginica	0	3	5	8
	10	10	10	30

Datos asociados a clases

	Accuracy	Recall	F-measure
Setosa	1	1	1
Versicolor	0.583	0.7	0.636
Virginica	0.625	0.5	0.556

F global=0.731

Matrices de confusión binarias

- Las clases ahora son P(positivos), N(negativos)
- La matriz de confusion es

Predichos\ciertos	Positivos	Negativos	
Positivos	TP	FP	$P_{pred} = TP + FP$
Negativos	FN	TN	$N_{pred} = FN + TN$
	$P_{ci} = TP + FN$	$N_{ci} = FP + TN$	n

- Se calculan las medidas según las expresiones anteriores.
- En el caso de que exista una gran descompensación entre casos se puede utilizar la matriz de costes

Evaluacion de modelos de clasificación

Problemas binarios: curvas ROC (Receiver Operating Characteristics)

Hipotesis de partida

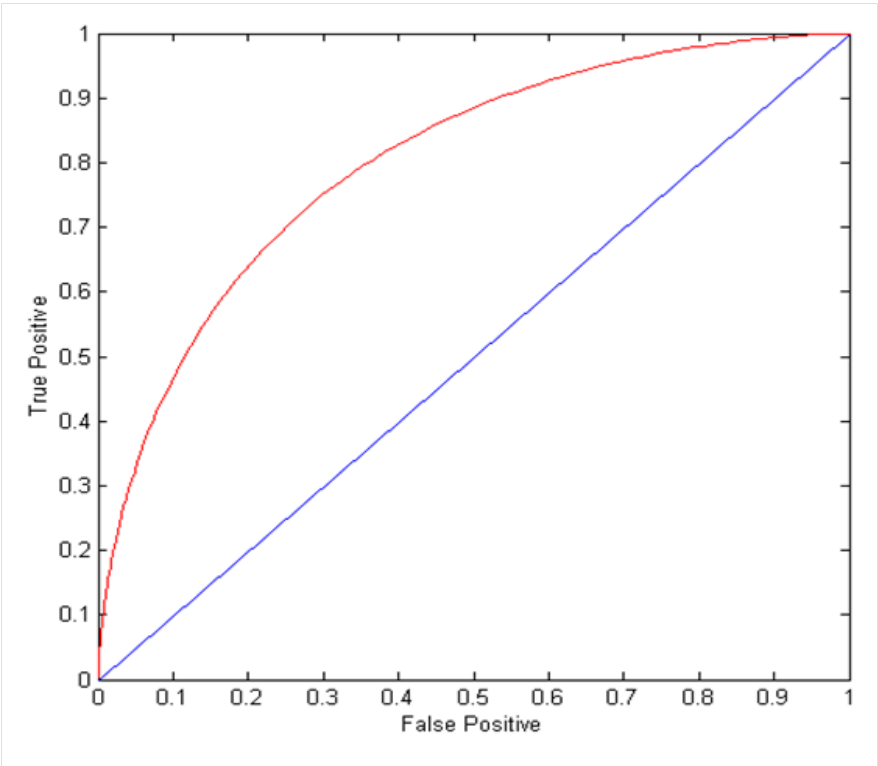
- Un problema binario
- Existe una medida $S(.)$ y un umbral ρ tal que si $S(x_i) > \rho$ se clasifica x_i como caso positivo. Por ejemplo, en un clasificador Bayes $S(x_i) = Prob(P|x_i)$ y sólo consideramos un ejemplo cómo positivo $S(x_i) > 0.8$

La curva ROC se construye haciendo variar el umbral ρ entre su valor mínimo y máximo y representando:

- En el eje Y la "Tasa de positivos ciertos" $TPR = \frac{TP}{P_c i} = \frac{TP}{(TP+FN)}$
- En el X la "Tasa de positivos falsos" $FPR = \frac{FP}{N_c i} = \frac{FP}{(FP+TN)}$

Evaluacion de modelos de clasificación

Problemas binarios: curvas ROC

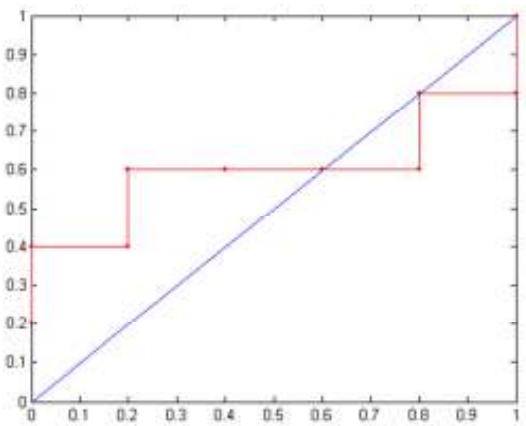


Si ρ es mínimo todos positivos (1,1)
Si ρ es máximo todos negativos
Igual número de falsos que de verdaderos
valor sobre la diagonal
Si la curva está por encima **Bueno**
Si la curva está por debajo **Malo**
Curva con valores cerca del (0,1) **lo mejor**
Area bajo la curva $\approx$ 1 perfecto

Evaluacion de modelos de clasificación

Problemas binarios: curvas ROC

Ejemplo de construcción



Ejemplo	P(+|E)	Clase
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

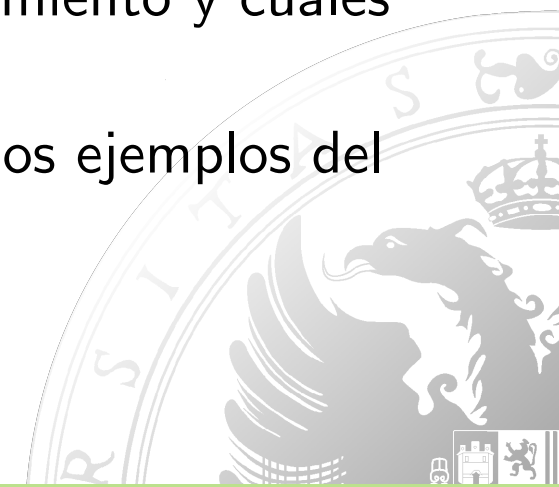
Class	+	-	+	-	-	-	+	-	+	+	
	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

88

Métodos para la evaluación de modelos de clasificación

Problema En un caso real, con un dataset determinado. ¿Cómo organizar los conjuntos de entrenamiento y test?

- Para evaluar la precisión de un modelo de clasificación, el conjunto test debe ser independiente
- Hay que dividir el dataset en dos. Por ejemplo, podríamos reservar 2/3 de los ejemplos disponibles para entrenamiento y el 1/3 restante lo utilizaríamos de conjunto test.
- *Problema* ¿Cómo dividir, qué datos van a entrenamiento y cuales al test?
- Una primera idea *Selección aleatoria*. Se sortean los ejemplos del conjunto test
  - Mediante sorteo uniforme global
  - Mediante sorteo estratificado según las clases



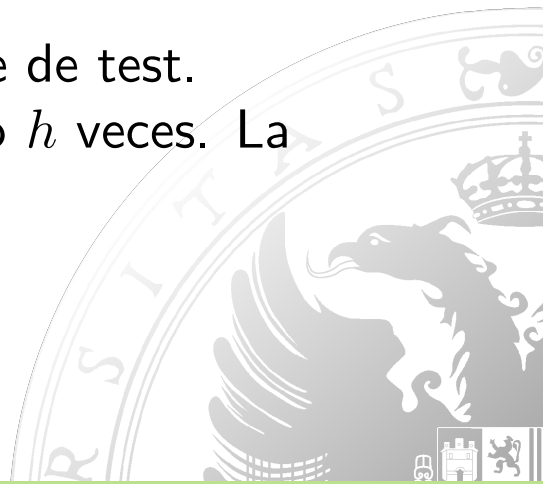
Métodos para la evaluación de modelos de clasificación

Validación cruzada

Problema

La selección aleatoria hecha sólo una vez puede estar sesgada

- Como solución se repite h veces el proceso y la precisión será la media $acc = 1/h \sum_{i=1}^{h} acc_i$
- Otra alternativa: *Validación cruzada*
  1. Se divide el dataset en h partes iguales
  2. Se cogen $h-1$ partes de entrenamiento y una parte de test.
  3. Se va variando la parte de test hasta repetir proceso h veces. La precisión es la media.



Métodos para la evaluación de modelos de clasificación

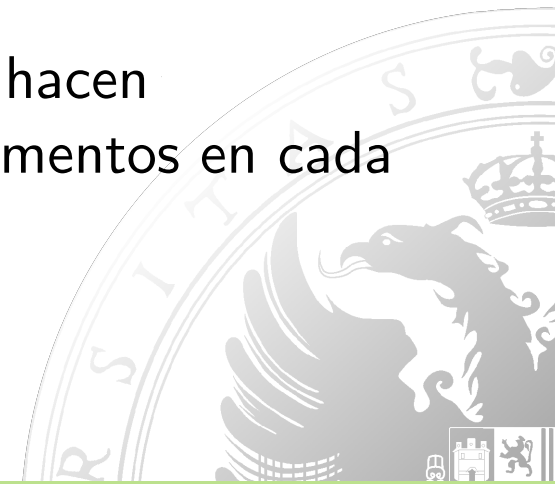
Validación cruzada

Variantes de la validación cruzada

Two-flod cross validation . En este caso h=2.

Leave-one-out . Si tenemos N ejemplos en el data set, dividimos N veces, dejando N-1 en entrenamiento y un caso en el test. Se realizan N ejecuciones del proceso de clasificación.

Validación cruzada estratificada . Las particiones se hacen manteniendo la proporción incial de elementos en cada clase



Métodos para la evaluación de modelos de clasificación

Bootrapping

- Se muestrea el conjunto de entrenamiento con reemplazamiento. Con lo que los ejemplos se pueden repetir
- Si N es suficientemente grande una muestra de tamaño N contiene aproximadamente el 63.2% de los ejemplos.
- Los datos que no se escojan forma parte del conjunto test
- El proceso se repite b veces con una precision de ε_i, $i = 1..b$.
- La precisión total se puede calcular de varias formas, la más habitual es:

$$acc_{boost} = 0.632(1/b)\sum_{i=1}^{b}\varepsilon_i + 0.368 acc_s$$

donde acc_s es la precisión obtenida utilizando el conjunto de entrenamiento total

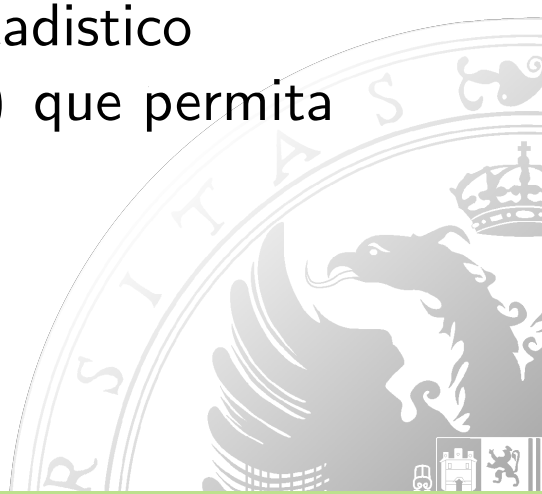
Comparación de clasificadores

Ideas básicas

Para comparar clasificadores:

Comparación genérica .

1. Se elige un conjunto de problemas
2. Se elige una medida de calidad. Habitualmente precisión.
3. Se evalúan y se realiza algún test estadistico (diferencia de medias, ANOVA etc.) que permita comparar resultados



Comparación de clasificadores

Ideas básicas

Para comparar clasificadores:

Comparación frente a problema concreto .

- Si los clasificadores son binarios se pueden utilizar curvas ROC para comparar su actuación sobre un problema concreto.
- Se pueden utilizar, validación cruzada o bootstrapping para generar experimentos y comparar resultados sin hacer medias.

No es fácil comparar clasificadores en general, lo más probable es que unos trabajen mejor que otros según la clase de problemas