Selección de modelos

Lección 15

Dr. Pablo Alvarado Moya

CE5506 Introducción al reconocimiento de patrones
Área de Ingeniería en Computadores
Tecnológico de Costa Rica

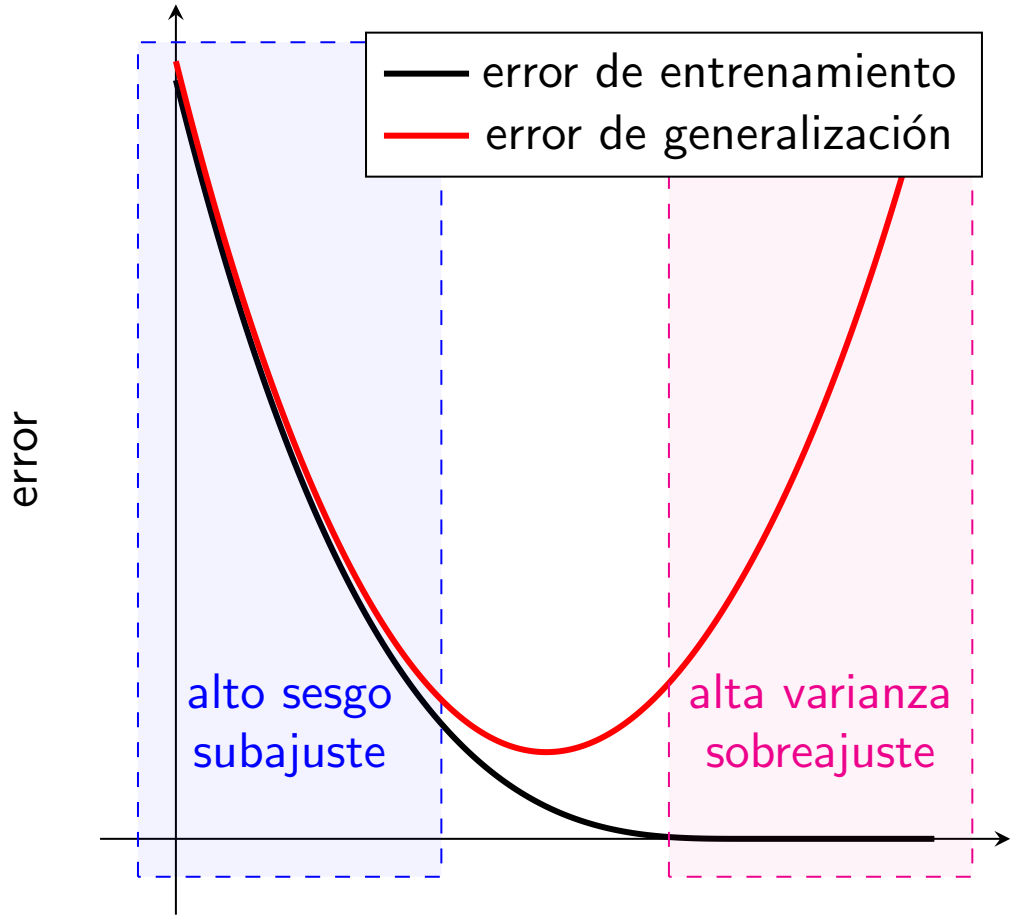
II Semestre, 2019

Contenido

1. Validación cruzada
2. Selección de características
3. Métricas de clasificación

Sesgo y varianza

- Vimos que existe un compromiso entre sesgo y varianza



Selección de modelo

- Querémos seleccionar en un modelo, por ejemplo,
  1. En regresión polinomial $h_{\underline{\boldsymbol{\theta}}}(\underline{\mathbf{x}}) = g(\theta_0 + \theta_1 x + \cdots + \theta_k x^k)$ seleccionar k
  2. En regresión ponderada localmente seleccionar τ
  3. En SVM seleccionar C
- Esto se conoce también como optimización de **hiperparámetros** (parámetros fijos durante el entrenamiento)

Selección ingenua

- Supongamos que tenemos un conjunto finito de modelos para elegir:

$$\mathcal{M} = \{M_1, \ldots, M_d\}$$

- Por ejemplo M_i puede ser un modelo polinomial de i-ésimo orden
- Los M_i pueden corresponder a distintos algoritmos (SVM, regresión logística, kNN, etc.)
- ¿Podríamos entrenar los d modelos y seleccionar el que dé el menor error? (¿por qué es esto una mala idea?)

Validación cruzada simple (primer intento)

- *Hold-out cross validation*
- Primero separe aleatoriamente el conjunto de entrenamiento $\mathcal{S}$ en $\mathcal{S}_{\text{train}}$ (digamos 70 % para entrenar) y $\mathcal{S}_{\text{cv}}$ (conjunto apartado para validación cruzada o *hold-out cross validation set*)
- Entrenar cada modelo M_i con $\mathcal{S}_{\text{train}}$ para obtener h_i
- Seleccionar la hipótesis h_i que produce el menor error $\hat{\varepsilon}_{\mathcal{S}_{\text{cv}}}(h_i)$ con el conjunto apartado
- La prueba con elementos no vistos durante el entrenamiento provee un mejor estimado del error de generalización
- Usualmente se usa de un 25 % a un 33 % como conjunto apartado, con 30 % el valor usual
- Si modelo ganador no es sensible a inicialización, puede entonces reentrenarse con $\mathcal{S}$

Estimación correcta de error de generalización

- El método simple tiene un problema de sesgo como estimador de error de generalización
- Como se selecciona el modelo M_i usando $\mathcal{S}_{\text{cv}}$, usar el error medido con $\mathcal{S}_{\text{cv}}$ como error de generalización tiende a ser optimista para M_i.
- Por ello se acostumbra a partir el conjunto de datos en **tres** (en vez de dos) subconjuntos: entrenamiento $\mathcal{S}_{\text{train}}$, validación cruzada $\mathcal{S}_{\text{cv}}$ y prueba $\mathcal{S}_{\text{test}}$.
- Con $\mathcal{S}_{\text{train}}$ se entrenan los modelos
- Con $\mathcal{S}_{\text{cv}}$ se seleccionan los modelos
- El conjunto $\mathcal{S}_{\text{test}}$ se usa para estimar el error de generalización final.

Validación cruzada de k iteraciones (1)

- El problema con la validación cruzada simple es que "desperdicia" los datos apartados.
- Esto no es problema si hay abundantes datos, pero en escenarios con pocos datos para entrenar, dejar datos fuera del entrenamiento es problemático.
- En estos casos de usa la validación cruzada de k iteraciones (*k-fold cross validation*)
- Los pasos de este método son los siguientes:
  1. Divida $\mathcal{S}$ aleatóreamente en k subconjuntos disjuntos de m/k datos de entrenamiento cada uno.
     Llamaremos a estos subconjuntos $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k$.

Validación cruzada de k iteraciones (2)

2. Evaluamos cada modelo M_i como sigue:
   Para $j = 1, \ldots, k$
   Entrene el modelo M_i con
   $\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_{j-1} \cup \mathcal{S}_{j+1} \cup \ldots \cup \mathcal{S}_k$.
   lo que da una hipótesis h_{ij}
   Pruebe hipótesis h_{ij} con $\mathcal{S}_j$ para obtener $\hat{\varepsilon}_{\mathcal{S}_j}(h_{ij})$
   El error de generalización de M_i se estima como el promedio para todo j de $\hat{\varepsilon}_{\mathcal{S}_j}(h_{ij})$
3. Tome el modelo M_i con el error de generalización estimado más bajo, y se reentrena con todo $\mathcal{S}$. La hipótesis resultante es la salida del método.

- Usualmente se utiliza $k = 10$
- En problemas críticos se llega a elegir $k = m$ lo que resulta en el método llamado validación cruzada dejando uno fuera (*leave-one-out*)

Selección de características

Selección de características

- Un caso particular de selección de modelos es el problema de **selección de características**
- Partamos de un problema de aprendizaje supervisado con un numero de características n elevado (quizá $n \gg m$)
- Posiblemente solo algunas de las características son relevantes
- Puesto que la dimensión VC de la hipótesis es $\mathcal{O}(n)$ tendremos sobreajuste, a menos que m sea suficientemente grande
- Lo que buscaremos es reducir n

Estrategia de selección de características

- Dadas n características, hay 2^n posibles subconjuntos de características
- El problema es entonces un problema de selección de modelo en donde se debe seleccionar uno de 2^n modelos
- El crecimiento exponencial de 2^n hace prohibitiva la búsqueda exhaustiva, en especial si se considera que el modelo subsiguiente también debe seleccionarse
- Por eso se usan selecciones aproximadas

Búsqueda hacia adelante

- La primera estrategia se denomina **búsqueda hacia adelante**

  Inicialice $\mathcal{F} = \emptyset$
  repeat
    foreach $i = 1, \ldots, n$ **do**
      if $i \notin \mathcal{F}$ **then**
        $\mathcal{F}_i = \mathcal{F} \cup \{i\}$
        Evalúe $\mathcal{F}_i$ con validación cruzada
      end
    end
    $\mathcal{F} \leftarrow \mathcal{F}_i$ con menor error
  until $|\mathcal{F}| > n_{\text{máx}}$
  Seleccione y retorne el mejor conjunto de características

Selección de características con envoltura

- La búsqueda hacia adelante es un caso particular de la **selección de características con envoltura de modelo** (*wrapper model feature selection*)
- En estos procesos se "envuelve" el algoritmo de aprendizaje que es llamado repetidas veces para probar distintas configuraciones de entrada.
- En la **búsqueda hacia atrás** se inicia con todas las características $\mathcal{F} = \{1, \ldots, n\}$ y se van borrando una por una.
- Estos métodos funcionan bien pero son caros por el número de veces que hay que entrenar el algoritmo
- ¡Búsqueda hacia adelante (o hacia atrás) tiene $\mathcal{O}(n^2)$ llamadas al algoritmo de entrenamiento!

Selección de características por filtrado

Filter feature selection

- En problemas con muchas características (como clasificación de texto), la selección con envoltura es demasiado costosa
- La **selección por filtrado** ignora al clasificador
- Se utilizan heurísticas rápidas de calcular para la selección
- La idea es calcular alguna puntuación (*score*) $S(i)$ sencilla que mida qué tan informativa es una característica x_i con respecto a las etiquetas y, para finalmente elegir las k características con mayor puntuación.
- Una posible medida es la correlación entre x_i e y

Información mutua

- En la práctica se usa con más frecuencia la **información mutua** $\mathrm{MI}(x_i,y)$, que para entradas binarias/discretas es

$$\mathrm{MI}(x_i,y) = \sum_{x_i \in \{0,1\}} \sum_{y \in \{0,1\}} p(x_i,y) \ln \frac{p(x_i,y)}{p(x_i)p(y)}$$

- Las probabilidades $p(x_i,y), p(x_i), p(y)$ se estiman empíricamente de los datos de entrenamiento
- MI es una medida conocida en teoría de la información, asociada con la entropía
- La información mútua equivale a la divergencia Kullback-Leibler (KL)

$$\mathrm{MI}(x_i,y) = \mathrm{KL}\left(p(x_i,y) \| p(x_i)p(y)\right)$$

Selección de k en filtrado

- En en el método por filtrado se seleccionan k características
- ¿Cómo seleccionamos k?
- Usualmente se utiliza correlación cruzada para distintos valores de k, y se selecciona el que tenga menor error

Métricas de clasificación

Métricas para evaluar clases desbalanceadas

- El caso de clases desbalanceadas (*skew classes*) requiere atención especial
- Ejemplo:
  Usted entrena un modelo de regresión logística $y = h_{\underline{\theta}}(\underline{\mathbf{x}})$
  - $y = 1$ si paciente tiene cáncer
  - $y = 0$ si paciente no tiene cáncer
- Usted obtiene un 1 % de error en conjunto de prueba (99 % de diagnósticos correctos)

Métricas para evaluar clases desbalanceadas

- El caso de clases desbalanceadas (*skew classes*) requiere atención especial
- Ejemplo:
  Usted entrena un modelo de regresión logística $y = h_{\underline{\theta}}(\underline{\mathbf{x}})$
  - $y = 1$ si paciente tiene cáncer
  - $y = 0$ si paciente no tiene cáncer
- Usted obtiene un 1 % de error en conjunto de prueba (99 % de diagnósticos correctos)
- En realidad solo 0,5 % de los pacientes tiene cáncer

Métricas para evaluar clases desbalanceadas

- El caso de clases desbalanceadas (*skew classes*) requiere atención especial
- Ejemplo:
  Usted entrena un modelo de regresión logística $y = h_{\underline{\theta}}(\underline{\mathbf{x}})$
  - $y = 1$ si paciente tiene cáncer
  - $y = 0$ si paciente no tiene cáncer
- Usted obtiene un 1 % de error en conjunto de prueba (99 % de diagnósticos correctos)
- En realidad solo 0,5 % de los pacientes tiene cáncer
- Si usáramos $h_{\underline{\theta}}(\underline{\mathbf{x}}) = 0$ tendríamos solo 0,5 % de error (¡menos que el clasificador!)

Clases desbalanceadas

- Si la tasa de ejemplos positivos a negativos es cercana a los extremos entonces se tienen clases **esviadas** o **desbalanceadas**
- La predicción constante es en estos casos aparenta ser "mejor".
- Necesitamos mejores métricas para medir estos casos.
- Supongamos que su algoritmo pasó de
  - 99,2 % de acierto (0,8 % error)
  - 99,5 % de acierto (0,5 % error)

  ¿es eso bueno?
- Usar en estos casos la tasa de aciertos no brinda información

Precisión contra exhaustividad

Precision/recall

- Precisión: o valor positivo predicho
- Exhaustividad: o sensibilidad o *recall*
- Valores derivados de la **matriz de confusión**
- En el caso binario:

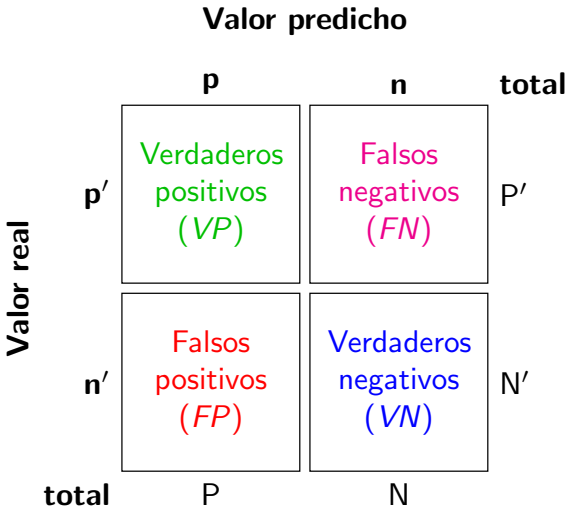
Valor real \ Valor predicho	p	n	total
p′	Verdaderos positivos (*VP*)	Falsos negativos (*FN*)	P′
n′	Falsos positivos (*FP*)	Verdaderos negativos (*VN*)	N′
total	P	N	

- Precisión: de todo lo que se predijo como $y = 1$, qué fracción realmente es 1
- Exhaustividad: de todos los pacientes con cáncer, qué fracción fue correctamente detectada

Precisión contra exhaustividad

Precision/recall

- Precisión: o valor positivo predicho
- Exhaustividad: o sensibilidad o *recall*
- Valores derivados de la **matriz de confusión**
- En el caso binario:



- Precisión: de todo lo que se predijo como $y = 1$, qué fracción realmente es 1

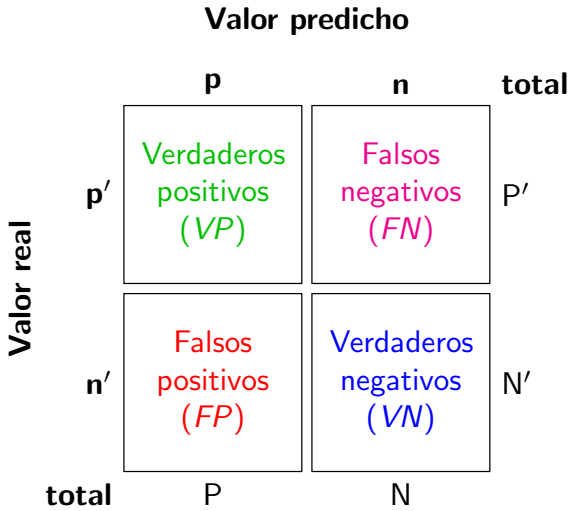
$$P = \frac{VP}{VP + FP}$$

- Exhaustividad: de todos los pacientes con cáncer, qué fracción fue correctamente detectada

Precisión contra exhaustividad

Precision/recall

- Precisión: o valor positivo predicho
- Exhaustividad: o sensibilidad o *recall*
- Valores derivados de la **matriz de confusión**
- En el caso binario:



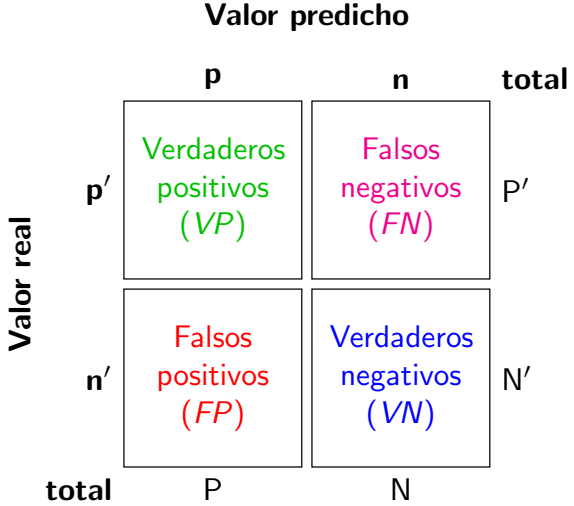
- Precisión: de todo lo que se predijo como $y = 1$, qué fracción realmente es 1
- Exhaustividad: de todos los pacientes con cáncer, qué fracción fue correctamente detectada

$$R = \frac{VP}{VP + FN}$$

Precisión contra exhaustividad

Precision/recall

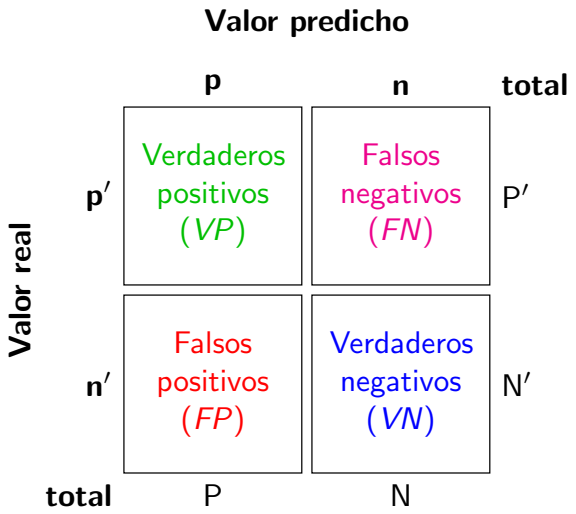
- Precisión: o valor positivo predicho
- Exhaustividad: o sensibilidad o *recall*
- Valores derivados de la **matriz de confusión**
- En el caso binario:



- Precisión: de todo lo que se predijo como $y = 1$, qué fracción realmente es 1
- Exhaustividad: de todos los pacientes con cáncer, qué fracción fue correctamente detectada

- La clase poco común corresponde a $y = 1$

Exactitud y error



- Precisión: $P = \dfrac{VP}{VP + FP}$
- Exhaustividad: $R = \dfrac{VP}{VP + FN}$
- Exactitud: $A = \dfrac{VP + VN}{VP + VN + FP + FN}$
- Error: $\varepsilon = \dfrac{FP + FN}{VP + VN + FP + FN}$

- Un buen clasificador tendrá ambos P y R altos
- El caso $h_{\underline{\theta}}(\underline{\mathbf{x}}) = 0$ tiene exhaustividad 0

Compromiso entre precisión y exhaustividad

- Retomemos nuestro clasificador con regresión logística
- $0 \leq h_{\underline{\boldsymbol{\theta}}}(\underline{\mathbf{x}}) \leq 1$
  - Predice 0 si $h_{\underline{\boldsymbol{\theta}}}(\underline{\mathbf{x}}) < 0{,}5$
  - Predice 1 si $h_{\underline{\boldsymbol{\theta}}}(\underline{\mathbf{x}}) \geq 0{,}5$

Compromiso entre precisión y exhaustividad

- Retomemos nuestro clasificador con regresión logística
- $0 \leq h_{\underline{\theta}}(\underline{\mathbf{x}}) \leq 1$
- **Caso 1**: Queremos predecir $y = 1$ (cáncer) únicamente si estamos muy seguros
  (no hay que asustar/dar tratamiento en vano)
  - Predice 0 si $h_{\underline{\theta}}(\underline{\mathbf{x}}) < \tau$
  - Predice 1 si $h_{\underline{\theta}}(\underline{\mathbf{x}}) \geq \tau$
  - Usamos $\tau = 0{,}7; 0{,}9; \ldots$
  - Se incrementa precisión, a costa de menor exhaustividad

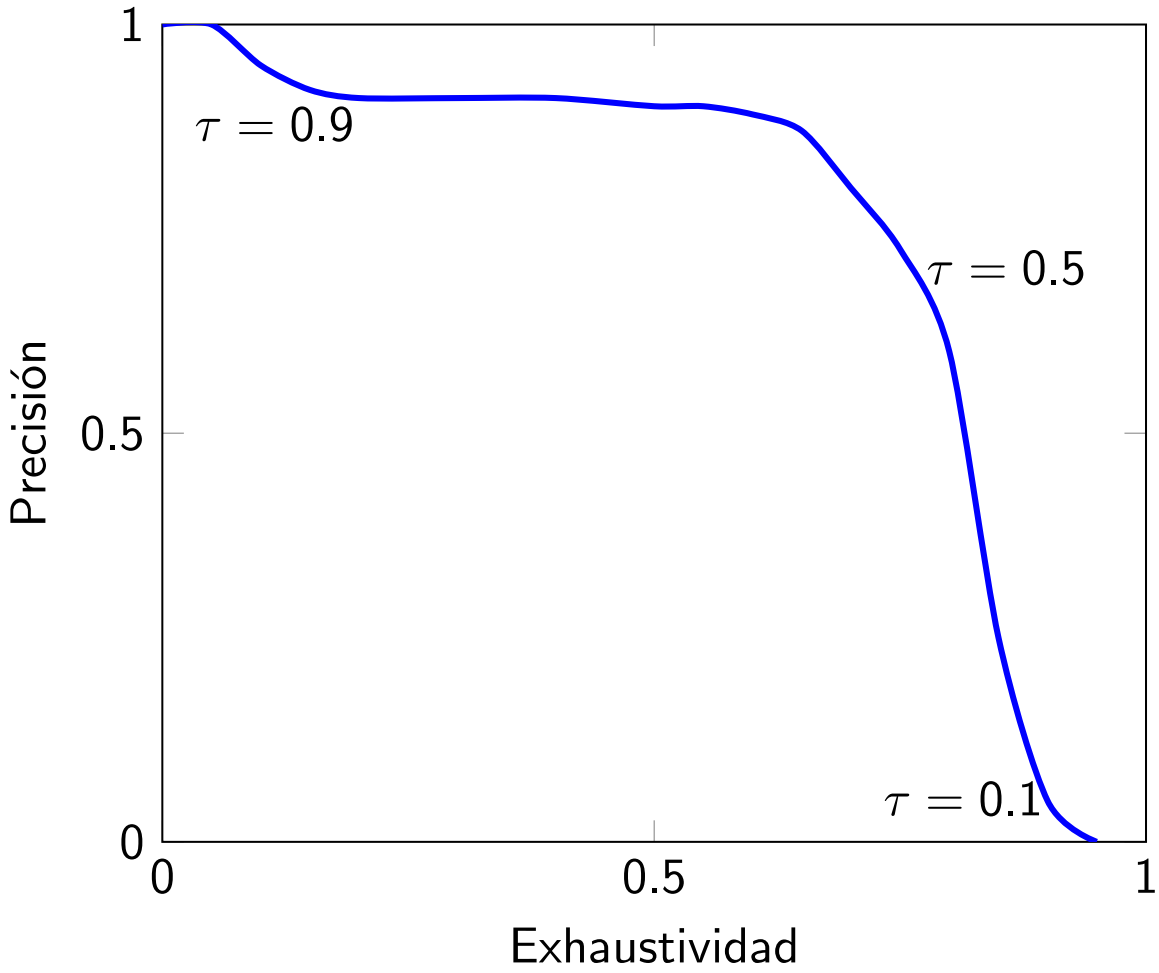
Compromiso entre precisión y exhaustividad

- Retomemos nuestro clasificador con regresión logística
- $0 \leq h_{\underline{\theta}}(\underline{\mathbf{x}}) \leq 1$
- **Caso 1**: Queremos predecir $y = 1$ (cáncer) únicamente si estamos muy seguros
- **Caso 2**: Queremos evitar pasar por alto cualquier caso de cáncer
  (no hay que omitir dar tratamiento a alguien que lo requiere)
  - Predice 0 si $h_{\underline{\theta}}(\underline{\mathbf{x}}) < \tau$
  - Predice 1 si $h_{\underline{\theta}}(\underline{\mathbf{x}}) \geq \tau$
  - Usamos $\tau = 0{,}4; 0{,}1; \ldots$
  - Se incrementa exhaustividad a costa de menor precisión

Compromiso entre precisión y exhaustividad

- Retomemos nuestro clasificador con regresión logística
- $0 \leq h_{\underline{\theta}}(\underline{\mathbf{x}}) \leq 1$
- **Caso 1**: Queremos predecir $y = 1$ (cáncer) únicamente si estamos muy seguros
- **Caso 2**: Queremos evitar pasar por alto cualquier caso de cáncer
- En general hay un compromiso en función de τ

Compromiso de precisión contra exhaustividad



Valor F

F_1 *score*

- ¿Cómo podemos seleccionar el valor de τ?
- ¿Cuál es un buen compromiso entre precisión y exhaustividad?
- Requerimos algún valor que evalúe el compromiso

Valor F

F_1 *score*

- ¿Cómo podemos seleccionar el valor de τ?
- ¿Cuál es un buen compromiso entre precisión y exhaustividad?
- Requerimos algún valor que evalúe el compromiso
- Promedio $(R + P)/2$ no funciona:

	P	R	$(R + P)/2$
Algoritmo 1	0,5	0,4	0,45
Algoritmo 2	0,7	0,1	0,4
Algoritmo 3	0,02	1	0,51

Valor F

F_1 *score*

- ¿Cómo podemos seleccionar el valor de τ?
- ¿Cuál es un buen compromiso entre precisión y exhaustividad?
- Requerimos algún valor que evalúe el compromiso

- Valor F (F_1 *score*) castiga los extremos:

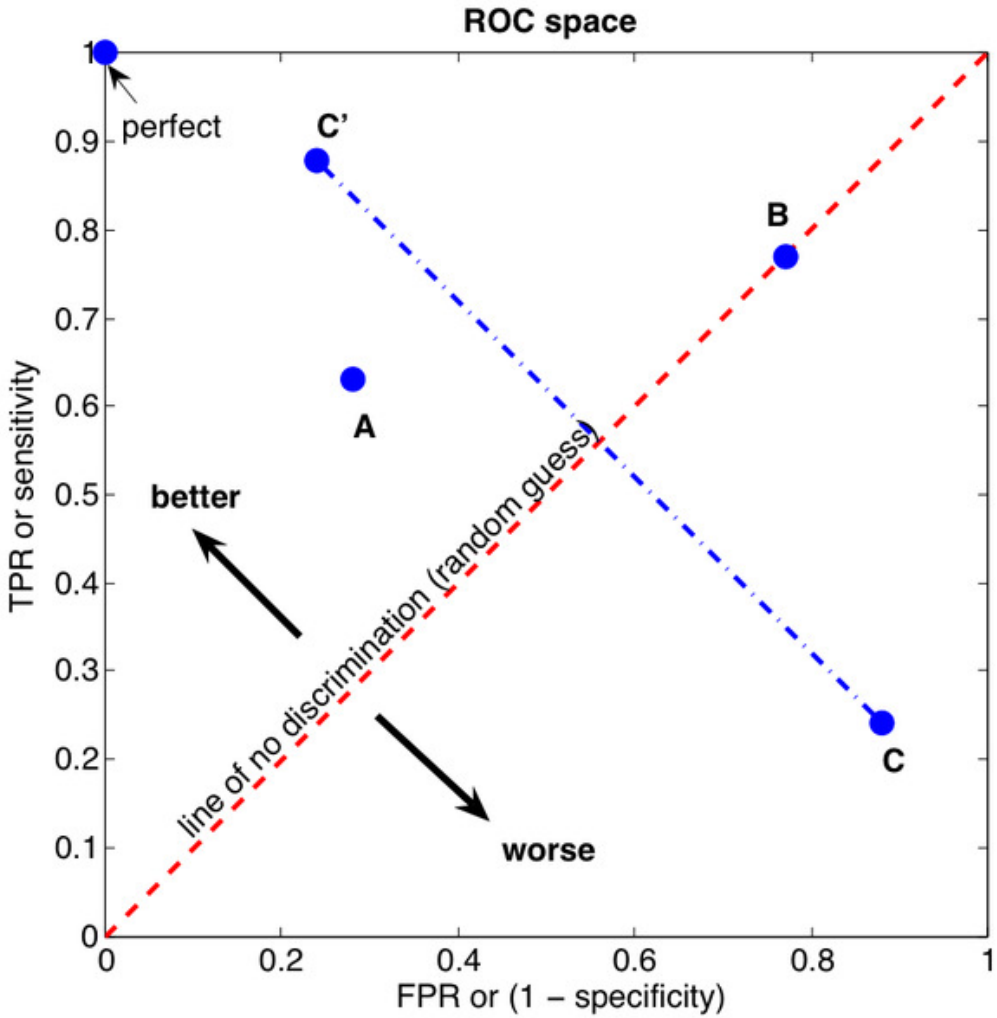
$$F_1 = 2\frac{PR}{P+R}$$

	P	R	$(R+P)/2$	F_1
Algoritmo 1	0,5	0,4	0,45	0,444
Algoritmo 2	0,7	0,1	0,4	0,175
Algoritmo 3	0,02	1	0,51	0,0392

Curvas ROC

- Las curvas *Receiver Operating Characteristic* también ofrecen una representación gráfica de la exhaustividad frente a la especificidad de un clasificador binario
  - Exhaustividad = VP/(VP+FN)
  - Especificidad = VN/(FP+VN)
- También se acostumbra utilizar la tasa de exhaustividad contra la tasa de falsos positivos
  - Exhaustividad = VP/(VP+FN)
  - Tasa de falsos positivos: FP/(FP+VN)

Curvas ROC


ROC space
perfect
C'
B
A
C
better
worse
line of no discrimination (random guess)
TPR or sensitivity
FPR or (1 – specificity)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.2
0.4
0.6
0.8
1

Matrices de confusión multiclase

Valor real \ Predicción	Clase A	Clase B	Clase C	Clase D	Clase E	Exhaustividad
Clase A	30	0	1	5	2	0.75
Clase B	2	35	1	0	2	0.875
Clase C	0	0	40	0	0	1.0
Clase D	1	0	0	39	0	0.97499
Clase E	5	4	2	4	25	0.625
Precisión	$\frac{30}{38}$	$\frac{35}{39}$	$\frac{40}{44}$	$\frac{39}{48}$	$\frac{25}{29}$	

Resumen

1. Validación cruzada
2. Selección de características
3. Métricas de clasificación

Este documento ha sido elaborado con software libre incluyendo LaTeX, Beamer, GNUPlot, GNU/Octave, XFig, Inkscape, GNU-Make y Subversion en GNU/Linux