Ricardo Aler Mur

EVALUACIÓN DE TÉCNICAS DE APRENDIZAJE-3

Evaluación y aprendizaje dependiente de la distribución y el coste

En esta clase se explica cómo evaluar modelos cuando la distribución de clases está desbalanceada y cuando los costes de clasificar erróneamente no son uniformes.

- Primero se considera la evaluación sensible a la distribución (clases desbalanceadas) y al coste (costes de errores no uniformes). Se explican las cuestiones que pueden surgir si insistimos en evaluar un problema desbalanceado (o con costes no uniformes) con el error de clasificación típico, y se introducen los conceptos de matriz de confusión y matriz de costes.
- Se introducen las ideas de contexto o condiciones operativas: matriz de costes + distribución de las clases y como computar el coste esperado
- Aunque el cálculo del coste esperado nos permite elegir el modelo más apropiado para determinadas condiciones operativas, puede ser más conveniente realizar un aprendizaje que optimice directamente dicho coste. Se habla entonces de aprendizaje sensible a la distribución y al Coste y se introducen los algoritmos SMOTE y Metacost

- Como Metacost está basado en la estimación de probabilidades usando Bagging, se habla del concepto de conjunto de clasificadores, Bagging y una técnica de Bagging específica para árboles: Random Forests.
- Se introduce la idea de curva ROC como una manera sistemática de analizar y seleccionar los clasificadores mas adecuados a un contexto determinado. Se explican las curvas ROC para clasificadores discretos y para scorers y la idea de convex hull, que permite descartar clasificadores ineficientes. Por último, se habla de la métrica AUC (área bajo la curva ROC) que es más adecuada para comparar clasificadores en problemas de muestra desbalanceada.
- Por último, se introducen las curvas de coste (cost curves), que es otra manera de representar gráficamente la relación entre el contexto y el coste de de un modelo.

CLASIFICACIÓN CON COSTES Y MUESTRAS DESBALANCEADAS

Organización

- Evaluación Sensible a la Distribución y al Coste
- Aprendizaje Sensible a la Distribución y al Coste (SMOTE, Metacost)
- Análisis ROC de Clasificadores discretos
- Análisis ROC de Scorers
- La Métrica AUC: el área bajo la curva ROC
- Curvas de coste (cost curves)

La Clasificación y su Evaluación

- Medida tradicional para evaluar clasificadores:
  - Error (o *accuracy*): porcentaje de instancias mal clasificadas (respecto al conjunto de test o utilizando validación cruzada).
  - En principio, un buen clasificador tiene que superar el error del azar:
    - En problemas de dos clases error<50% (o accuracy > 50%)
    - En problemas de m clases, error<100/m

EVALUACIÓN SENSIBLE A LA DISTRIBUCIÓN Y EL COSTE

- Evaluación Sensible a la Distribución:
  - No siempre todas las clases tienen la misma proporción (no están balanceadas)
- Ejemplo:
  - Problema con personas enfermas / sanas:
    - Cáncer: 1%
    - No Cáncer: 99%
  - Si un clasificador trivial predice siempre “No cáncer” tendrá un error del 1% (muy pequeño), pero predice incorrectamente de manera sistemática a los que tienen cáncer
  - Para tener en cuenta una distribución desbalanceada a la hora de evaluar un clasificador, el porcentaje de acierto debería ser mayor, al menos, que el porcentaje de datos de la clase mayoritaria (de otra manera, para el clasificador. Es decir, el clasificador tiene que ser mejor que el clasificador trivial.

EVALUACIÓN SENSIBLE A LA DISTRIBUCIÓN Y EL COSTE

Matriz de confusión: se puede utilizar para desglosar los porcentajes de acierto por cada clase

		Real	
	CÁNCER	si(p) pos=TP+FN	no (n) neg=FP+TN
Predicho	SI (P)	TPR=TP/pos	FPR=FP/neg
	NO (N)	FNR=FN/pos	TNR=TN/neg

EVALUACIÓN SENSIBLE A LA DISTRIBUCIÓN Y EL COSTE

¿Cuál es el mejor clasificador?. Nótese que ambos tienen el mismo porcentaje de aciertos (90+60)/200

Podemos utilizar la matriz de confusión para quedarnos con el mejor clasificador, que en este caso es el que minimiza los FN (el de arriba), puesto que lo peor es decirle que no tiene cáncer a una persona que realmente lo tiene

CÁNCER	si(p)	no (n)
SI (P)	TP=90	FP=40
NO (N)	FN=10	TN=60

CÁNCER	si(p)	no (n)
SI (P)	TP=60	FP=10
NO (N)	FN=40	TN=90

EVALUACIÓN SENSIBLE A LA DISTRIBUCIÓN Y EL COSTE

- Evaluación sensible al coste:
  - Ejemplo: en el caso de cáncer / no cáncer no es lo mismo decirle que no tiene cáncer a una persona que si que lo tiene, que el error contrario.
  - Nòtese que esta es una situación diferente a la de distribución desbalanceada
  - Matriz de costes:

		Real	
	CÁNCER	si	no
Predicho	SI	0	10€
	NO	10000€	0

  - Lo importante no es obtener un clasificador que yerre lo menos posible sino que tenga el mínimo coste. Podemos generar varios clasificadores, calcular sus matrices de confusión, y utilizar la matriz de costes para evaluar el coste final. Nos quedaremos con aquel que tenga menos coste

ORGANIZACIÓN

- **Evaluación Sensible** a la Distribución y **al Coste**
- Aprendizaje Sensible a la Distribución y al Coste (SMOTE, Metacost)
- Análisis ROC de Clasificadores discretos
- Análisis ROC de Scorers
- La Métrica AUC: el área bajo la curva ROC
- Curvas de coste (cost curves)

EVALUACIÓN SENSIBLE A LA DISTRIBUCIÓN Y EL COSTE

- Definición: **contexto o condiciones operativas**:
  - Distribución de instancias
    - Ej: 90% pertenecen a la clase positiva, 10% a la clase negativa
    - Pos = 0.9, Neg = 0.1
  - Matriz de costes

Real

Predicho	CÁNCER	si	no
	SI	0	10€
	NO	10000€	0

Real

Predicho	CÁNCER	si	no
	SI	CostTP	CostFP
	NO	CostFN	CostTN

EVALUACIÓN SENSIBLE A LA DISTRIBUCIÓN Y EL COSTE

- Definición: **coste esperado**:
  - El coste de los aciertos es cero

  Coste esperado =

  Número de falsos negativos * coste falsos negativos +

  + Número de falsos positivos * coste falsos positivos

  - Recordar que:
    - FNR = FN / Numero de positivos
    - FPR = FP / Número de negativos

Predicho \ Real	si(p) pos=TP+FN	no(n) neg=FP+TN
SI (P)	TPR=TP/pos	FPR=FP/neg
NO (N)	FNR=FN/pos	TNR=TN/neg

Coste = falsos negativos* CostFN + falsos positivos * CostFP

Coste = Pos * FNR * CostFN + Neg * FPR * CostFP

EVALUACIÓN SENSIBLE A LA DISTRIBUCIÓN Y EL COSTE

- Definición: **coste esperado**:si

Coste = Pos * FNR * CostFN + Neg * FPR * CostFP

Real

Predicho

	si	no
SI	0	10€
NO	10000€	0

Matriz de confusión

	si (p)	no (n)
SI (P)	TPR	FPR
NO (N)	FNR	TNR

Real

Predicho

	si	no
SI	CostTP	CostFP
NO	CostFN	CostTN

Coste = Pos * FNR * CostFN + Neg * CostFP * FPR

EVALUACIÓN SENSIBLE A LA DISTRIBUCIÓN Y EL COSTE

- Definición: **coste esperado**:

Coste = Pos * FNR * CostFN + Neg * CostFP * FPR

  - Se calcula a partir de

    - Distribución de instancias Pos, Neg
    - Matriz de costes

Real

Predicho	si	no
SI	0	10€
NO	10000€	0

Matriz de confusión

	si (p)	no (n)
SI (P)	TPR	FPR
NO (N)	FNR	TNR

Real

Predicho	si	no
SI	CostTP	CostFP
NO	CostFN	CostTN

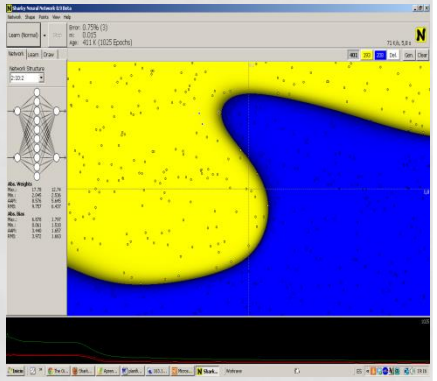
Coste = Pos * FNR * CostFN + Neg * CostFP * FPR

ORGANIZACIÓN

- Evaluación Sensible a la Distribución y al Coste
- **Aprendizaje Sensible a la Distribución** y al Coste (SMOTE, Metacost)
- Análisis ROC de Clasificadores discretos
- Análisis ROC de Scorers
- La Métrica AUC: el área bajo la curva ROC
- Curvas de coste (cost curves)

RECORDATORIO: SCORERS, ESTIMACIÓN DE PROBABILIDADES

- *Clasificación discreta:* $g : X \rightarrow \{0,1\}$
- *Scoring: en caso de que* la función g nos devuelva una medida de pertenencia de una instancia a una clase:
  - $g : X \rightarrow \mathbf{R}$ (en caso de clasificación binaria). Por ejemplo, *g(x)* = distancia de x a la frontera de separación
    - Si el valor es muy negativo, cercano a la clase 0
    - Si el valor es muy positivo, cercano a la clase 1
    - También podemos normalizar a [0,1]: $g : X \rightarrow [0,1]$
- Estimación de probabilidades:
  - Un "score" no es una probabilidad (no cumple las leyes de las probabilidades)
  - g(x) = p(y|x)
    - Con tres clases 0, 1 y 2: p(y=0|x), p(y=1|x), p(y=2|x)

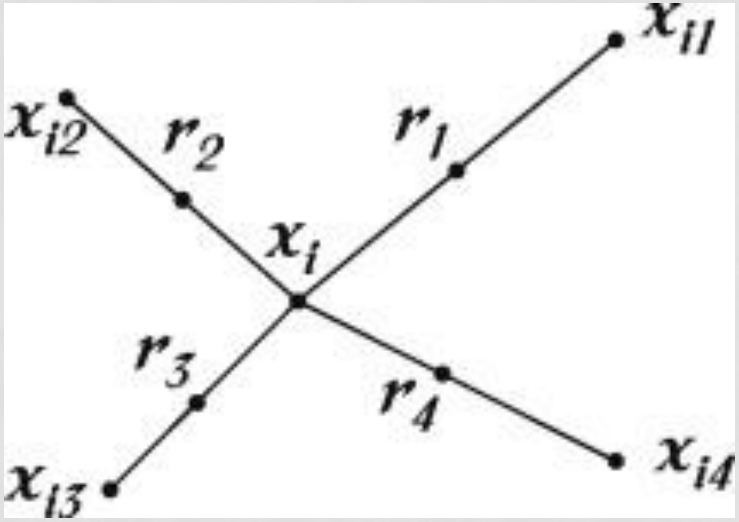


APRENDIZAJE SENSIBLE A LA DISTRIBUCIÓN

- Los algoritmos de aprendizaje típicos tienden a maximizar el porcentaje de aciertos
- En problemas de muestra desbalanceada, suele equivaler a aprender bien la clase mayoritaria a costa de aprender mal la minoritaria (recordar ejemplo cáncer)
- Solución 1: aprender varios modelos (p. ej. con distintos algoritmos) y seleccionar aquel que aprenda bien la clase minoritaria
- Solución 2: remuestreo:
  - Submuestreo: eliminar datos de la clase mayoritaria para equilibrarla con la minoritaria
  - Sobremuestreo: replicar datos de la clase minoritaria

APRENDIZAJE SENSIBLE A LA DISTRIBUCIÓN: SMOTE

- SMOTE: Synthetic Minority Over-sampling Technique:
  - Se generan instancias entre instancias de la clase minoritaria (se consideran k vecinos)
  - Se suele utilizar la regla de edición de Wilson para eliminar datos ruido



Fuente: Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2011). SMOTE: synthetic minority over-sampling technique. *arXiv preprint arXiv:1106.1813*.

ORGANIZACIÓN

- Evaluación Sensible a la Distribución y al Coste
- **Aprendizaje Sensible** a la Distribución y **al Coste** (SMOTE, Metacost)
- Análisis ROC de Clasificadores discretos
- Análisis ROC de Scorers
- La Métrica AUC: el área bajo la curva ROC
- Curvas de coste (cost curves)

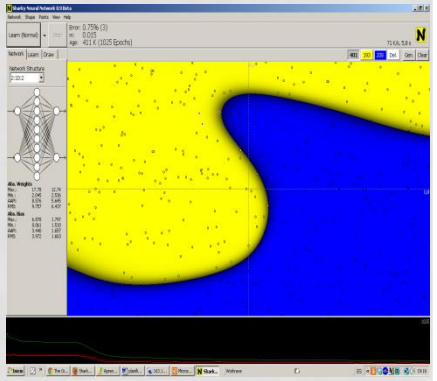
APRENDIZAJE SENSIBLE AL COSTE

- La mayor parte de los algoritmos de generación de clasificadores asumen implícitamente igualdad de costes y la distribución presente en los datos de entrenamiento
- Eso quiere decir que si los costes son distintos, el clasificador puede no ser óptimo desde el punto de vista del coste (aunque lo sea desde el punto de vista del porcentaje de aciertos)
- ¿Cómo aprender un clasificador que sea óptimo para una matriz de costes concreta?

		Real	
		si	no
Predicho	SI	0	10€
	NO	10000€	0

RECORDATORIO: SCORERS, ESTIMACIÓN DE PROBABILIDADES

- ***Clasificación discreta***: $g : X \rightarrow \{0,1\}$
- ***Scoring***: *en caso de que* la función g nos devuelva una medida de pertenencia de una instancia a una clase:
  - $g : X \rightarrow \boldsymbol{R}$ (en caso de clasificación binaria). Por ejemplo, *g(x)* = distancia de x a la frontera de separación
    - Si el valor es muy negativo, cercano a la clase 0
    - Si el valor es muy positivo, cercano a la clase 1
    - También podemos normalizar a [0,1]: $g : X \rightarrow [0,1]$
- **Estimación de probabilidades:**
  - Un "score" no es una probabilidad (no cumple las leyes de las probabilidades)
  - g(x) = p(y | x)
    - Con tres clases 0, 1 y 2: p(y=0 | x), p(y=1 | x), p(y=2 | x)

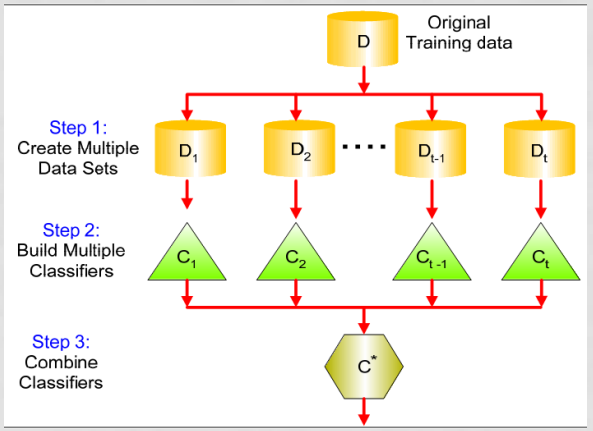


APRENDIZAJE SENSIBLE AL COSTE: METACOST

- Si nuestro clasificador fuera un estimador de probabilidades, sería fácil. Ejemplo:
- $Coste_x(pos) = CostFP*P(C=neg \mid x)$
  - CostFN / CostFP = coste de falsos negativos y falsos positivos
  - Asumimos que los aciertos tienen coste cero
- $Coste_x(neg) = CostFN*P(C=pos \mid x)$
- Y nos quedamos con la clase que tenga el coste mínimo
- Pero la mayor parte de los algoritmos que generan clasificadores no son estimadores de probabilidades.
- Solución: Bagging
- Algoritmos en Weka que siguen esta solución, los metaclasificadores:
  - Metacost
  - Cost-sensitive classifier

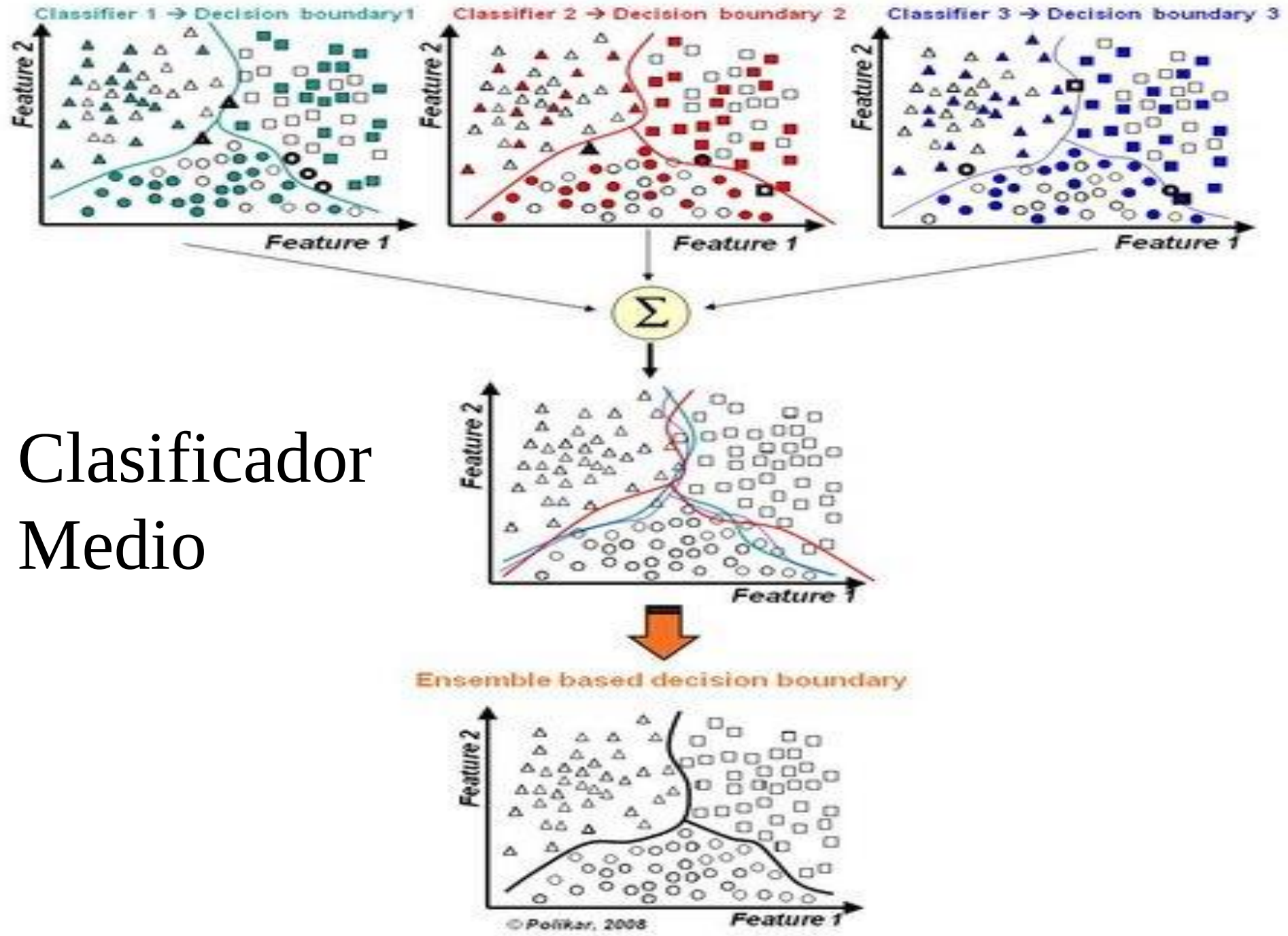
APRENDIZAJE SENSIBLE AL COSTE: BAGGING

- A partir del conjunto de entrenamiento, genera varios conjuntos de entrenamiento mediante remuestreo con reemplazamiento
- Para cada conjunto aprende un clasificador distinto
- Para estimar las probabilidades *p(Clase | x)* de una instancia, contar el número de clasificadores que la predicen como positiva y el número que la predicen como negativa



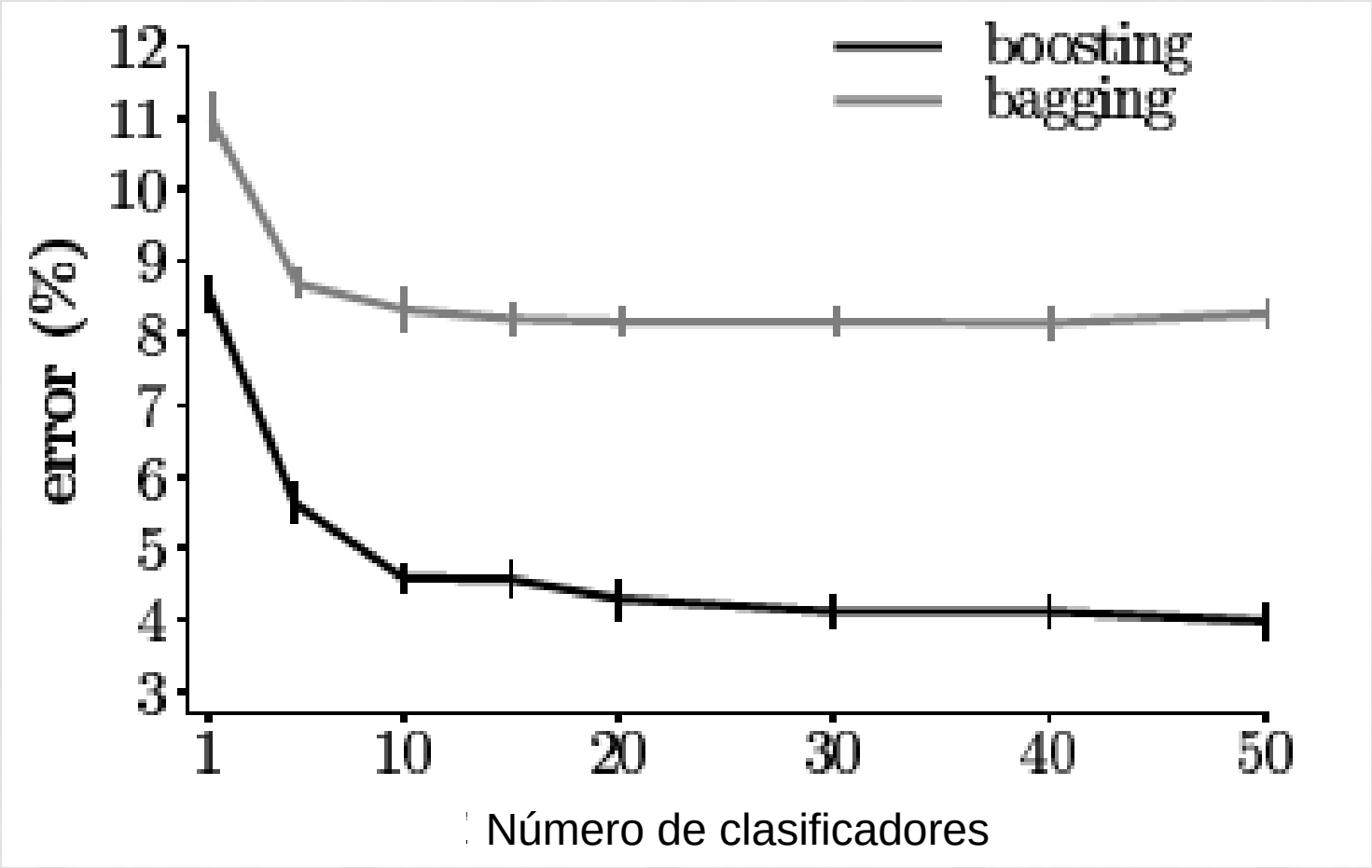
BAGGING (BOOTSTRAP AGGREGATING)

- Justificación: un algoritmo de aprendizaje automático genera clasificadores distintos si le pasamos datos de entrenamiento distintos
- Si el algoritmo es inestable, pequeñas diferencias en los datos de entrenamiento darán lugar a clasificadores muy distintos
  - Inestables: árboles de decisión, árboles de regresión, decision stumps (árboles con un solo nodo), redes de neuronas, …
  - Estables: vecino más cercano (IB1, IBK), máquinas de vectores de soporte (SMO, …)
- Solución: generar muchos conjuntos de entrenamiento y entrenar con cada uno de ellos un clasificador. La clase del clasificador agregado se decidirá por votación
- Los diferentes conjuntos de entrenamiento se generan a partir del conjunto de entrenamiento original por medio de muestreo aleatorio.



Clasificador Medio

BAGGING Y DESCENSO DEL ERROR



RANDOMIZATION

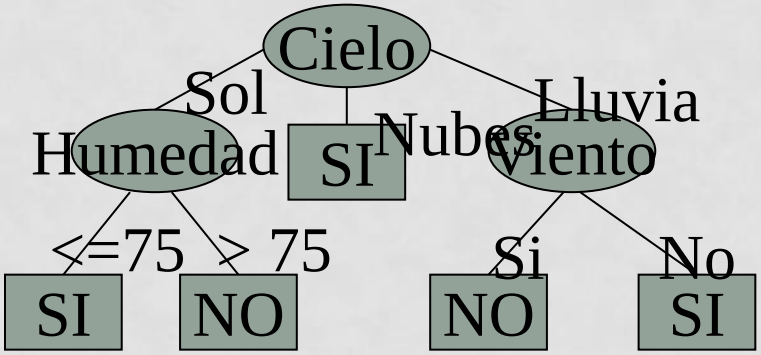
- Nota: también se pueden crear conjuntos de clasificadores generando distintos clasificadores a partir del mismo conjunto de entrenamiento, mediante randomización
- Ej: en redes de neuronas, como los pesos iniciales se inicializan aleatoriamente, diferentes procesos de aprendizaje generarán distintas redes incluso a partir del mismo conjunto de datos disponibles

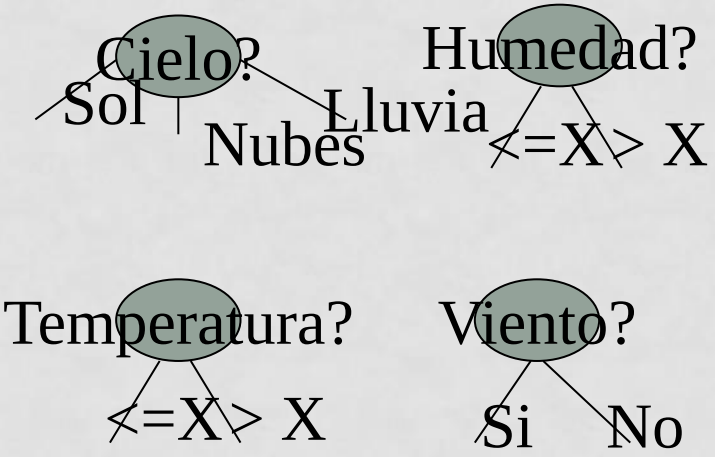
CONJUNTOS (ENSEMBLES) DE CLASIFICADORES

- Tipos principales:
  - Bagging:
    - **Random Forests**: Crea un ensemble de varios árboles de decisión
  - Boosting:
  - Stacking:

RANDOM FORESTS

- Es Bagging con árboles de decisión (por ejemplo, creados con J48 o C4.5)
- Usa dos métodos de randomización:
  - Genera varias submuestras de datos (como Bagging estándar)
  - En cada nodo de cada árbol pone, no el mejor atributo de entre todos los disponibles, sino el mejor elegido de entre **m** seleccionados aleatoriamente


Cielo
Sol
Nubes
Lluvia
Humedad
SI
Viento
<=75
> 75
Si
No
SI
NO
NO
SI

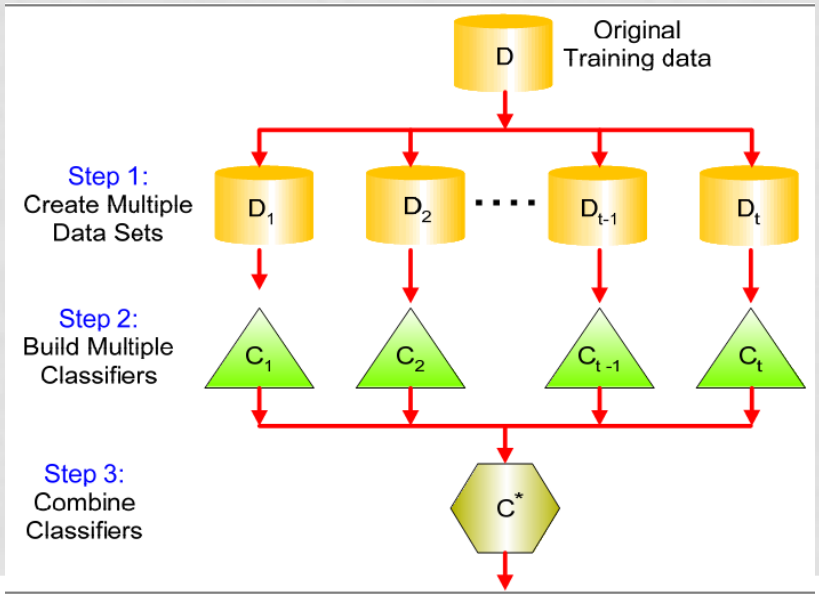

Cielo?
Sol
Nubes
Lluvia
Humedad?
<=X
> X
Temperatura?
<=X
> X
Viento?
Si
No

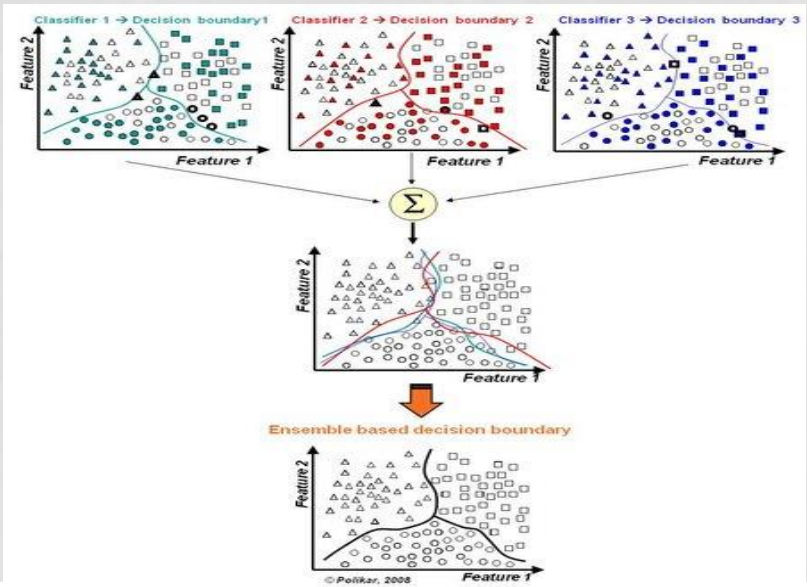
RANDOM FORESTS

- Sólo dos parámetros: número *k* de árboles en el ensemble y número *m* de atributos para ser tenidos en cuenta en cada creación de nodo
- Los Random Forests es de los algoritmos que mejor funcionan en clasificación y estimación de probabilidades

APRENDIZAJE SENSIBLE AL COSTE

- Para estimar las probabilidades *p(Clase | x)* de una instancia, contar el número de clasificadores que la predicen como positiva y el número que la predicen como negativa
- Es sólo una estimación, pero tiene sentido porque aquellos datos que estén cerca de la frontera tendrán probabilidades más cercanas al 50% (la mitad de los árboles están en desacuerdo) y aquellos datos más lejanos, más cercanas al 100% (todos los árboles de acuerdo)





ORGANIZACIÓN

- Evaluación Sensible a la Distribución y al Coste
- Aprendizaje Sensible a la Distribución y al Coste (SMOTE, Metacost)
- **Análisis ROC de Clasificadores discretos**
- Análisis ROC de Scorers
- La Métrica AUC: el área bajo la curva ROC
- Curvas de coste (cost curves)

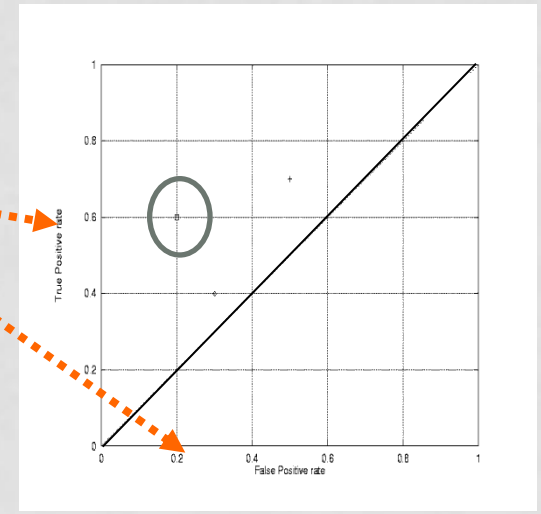
ANÁLISIS ROC DE CLASIFICADORES DISCRETOS

- Hemos visto que si conocemos la distribución de las clases y/o la matriz de costes (contexto o condiciones operativas), podemos aprender un clasificador optimizado para esas condiciones
- PROBLEMA:
  - En muchas aplicaciones, *hasta el momento de aplicación,* no se conoce el contexto. P.ej. un clasificador de spam.
  - Por tanto, es necesario realizar el aprendizaje del clasificador en un contexto (el de los datos de entrenamiento) distinto al de futuro uso.
  - Una posibilidad es aprender el clasificador, pero adaptarlo al contexto futuro una vez que se conozca.
- **Análisis ROC** (*Receiver Operating Characteristic*): una curva ROC es una representación gráfica del funcionamiento de un clasificador para todos los contextos posibles

ANÁLISIS ROC DE CLASIFICADORES DISCRETOS

- El espacio ROC
  - Se normaliza la matriz de confusión por columnas y calcular TPR, FNR TNR, FPR. Se representa en el espacio ROC con las coordenadas (X=FPR, Y=TPR)

		Real	
		si	no
Pred	SI	**TPR=0,8**	**FPR=0,2**
	NO	**FNR=0,2**	**TNR=0,8**



¿PORQUÉ LA DIAGONAL ES ALEATORIA?

- TPR = FPR
- Supongamos que un clasificador clasifica **aleatoriamente** los datos como:
  - positivo el 50% de las veces
  - negativo el 50% de las veces
  - Por pura casualidad, acertará con el 50% de los positivos y fallará con el 50% de los negativos
    - TPR = 0.5; FPR=0.5

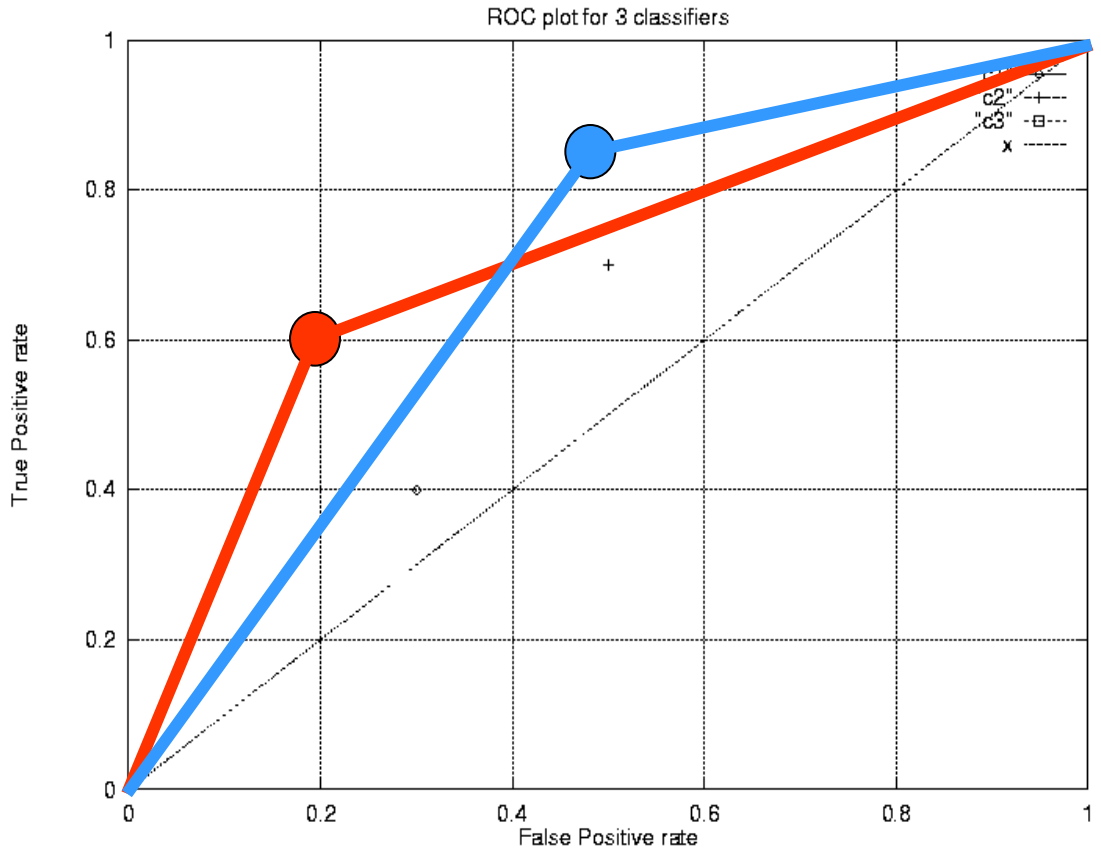
¿PORQUÉ LA DIAGONAL ES ALEATORIA?

- TPR = FPR
- Supongamos que un clasificador clasifica **aleatoriamente** los datos como:
  - positivo el 90% de las veces
  - negativo el 10% de las veces
  - Por pura casualidad, acertará con el 90% de los positivos y fallará con el 90% de los negativos
    - TPR = 0.9; FPR=0.9

ANÁLISIS ROC DE CLASIFICADORES DISCRETOS

- ¿Y si tenemos varios clasificadores discretos?
- Algunos funcionarán bien en un rango de contextos y otros en otros

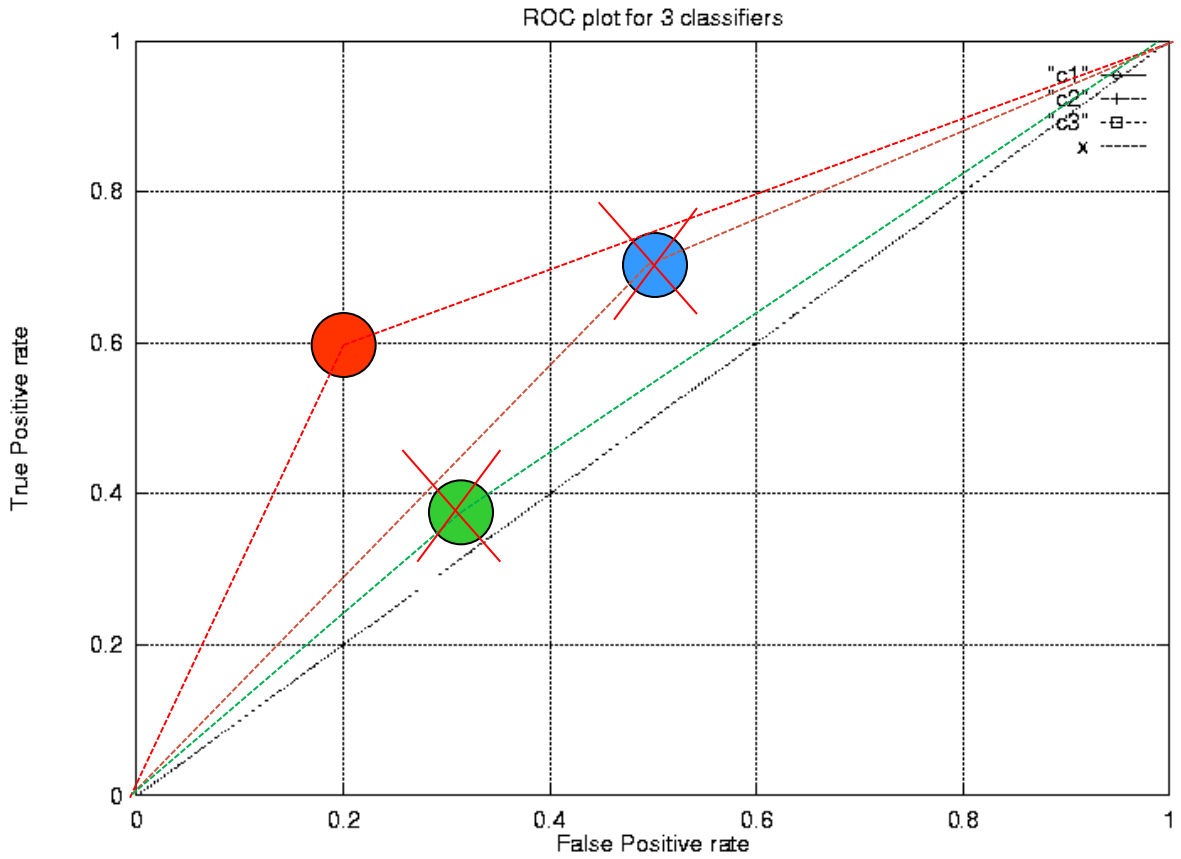
COMPARACIÓN DE CLASIFICADORES DISCRETOS



ANÁLISIS ROC DE CLASIFICADORES DISCRETOS

- ¿Y si tenemos varios clasificadores discretos?
- Algunos funcionarán bien en un rango de contextos y otros en otros
- ¿Es posible determinar si algunos clasificadores funcionar peor en todos los contextos y por tanto son descartables?

DOMINANCIA EN CURVAS ROC



El clasificador rojo domina al azul y al verde PARA TODOS LOS POSIBLES CONTEXTOS (o condiciones operativas)

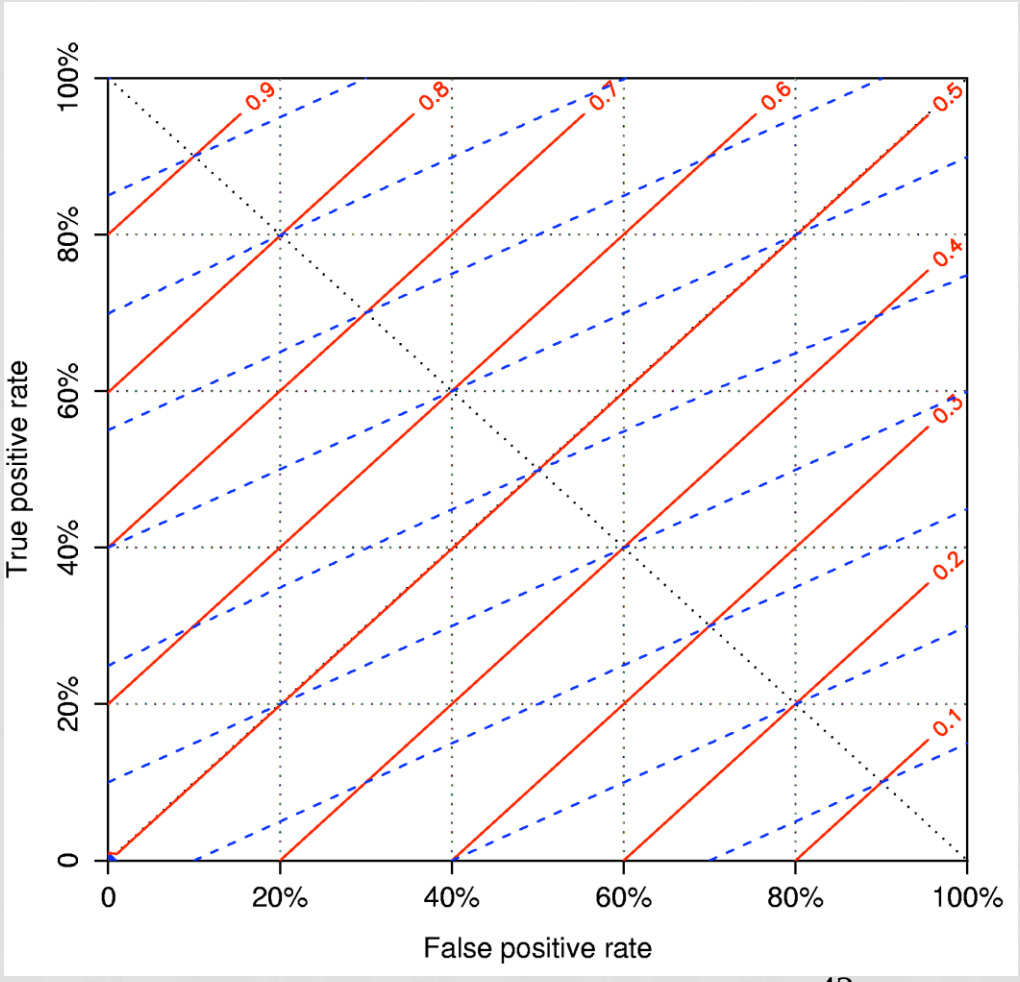
LÍNEAS DE ISOCOSTE

- Coste medio = Pos*CostFN*FNR+Neg*CostFP*FPR
  - Sea PCN = Pos*CostFN
  - Sea NCP = Neg*CostFP
  - TPR + FNR = 1
    - (los positivos que se clasifican bien mas los que se clasifican mal suman todos los positivos)
- Coste = PCN*FNR+NCP*FPR = PCN*(1-TPR)+NCP*FPR
- Coste = NCP*FPR-PCN*TPR+PCN

LÍNEAS DE ISOCOSTE

- Coste = NCP*FPR-PCN*TPR+PCN
- Todos los puntos (FPR,TPR) cuyo coste es una constante k forman una línea:
  - NCP*FPR-PCN*TPR+PCN = k
  - TPR = ((k-PCN)-NCP*FPR)/(-PCN)
  - TPR = ((k-PCN)-NCP*FPR)/(-PCN)
  - TPR = a*FPR+b = slope * FPR + b
    - a = NCP/PCN = (Neg * CostFP) / (Pos * CostFN)
    - b = (PCN-k)/PCN = 1-k/PCN
  - y=a*x+b

LÍNEAS DE ISOCOSTE



y=a*x+b

- a = NCP/PCN = (Neg * CostFP) / (Pos * CostFN)
-
- b = 1-k/PCN

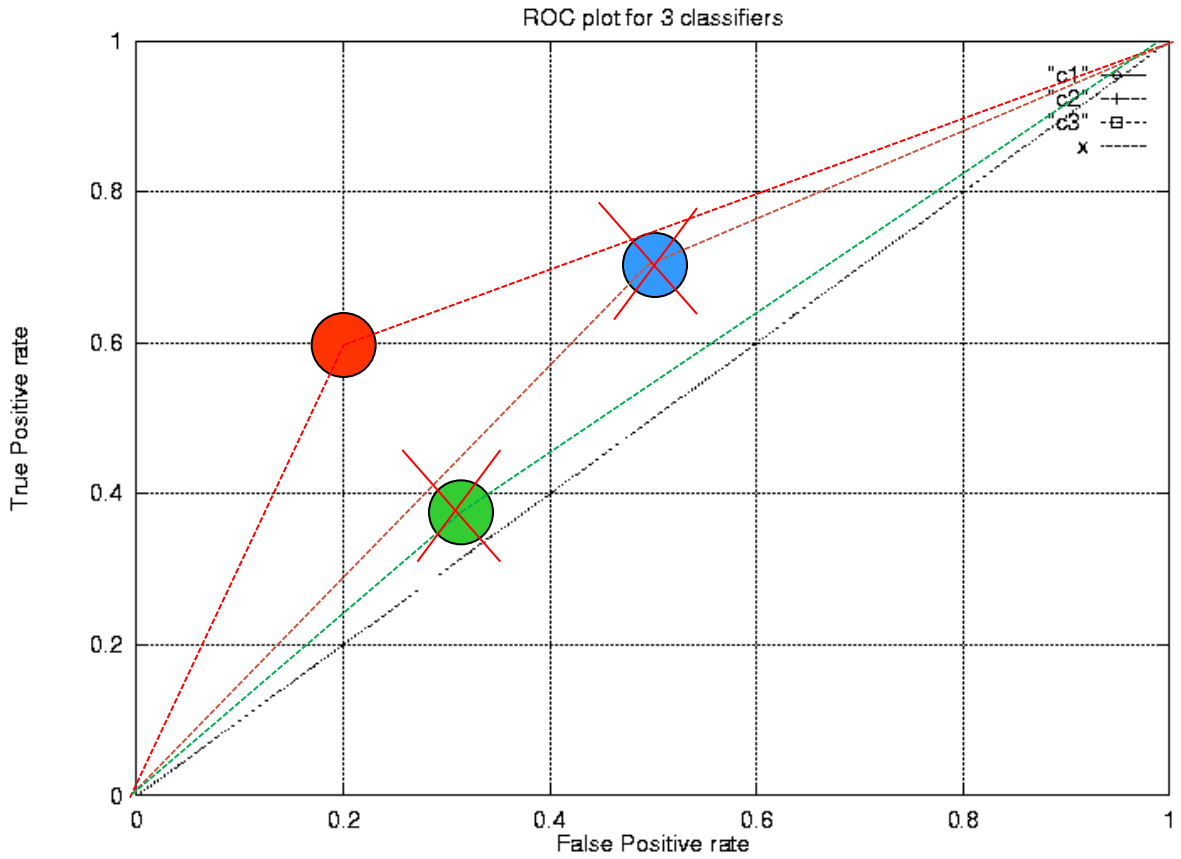
Si x=0, y = b = 1-k/PCN

Las líneas paralelas representan costes crecientes.

LÍNEAS DE ISOCOSTE

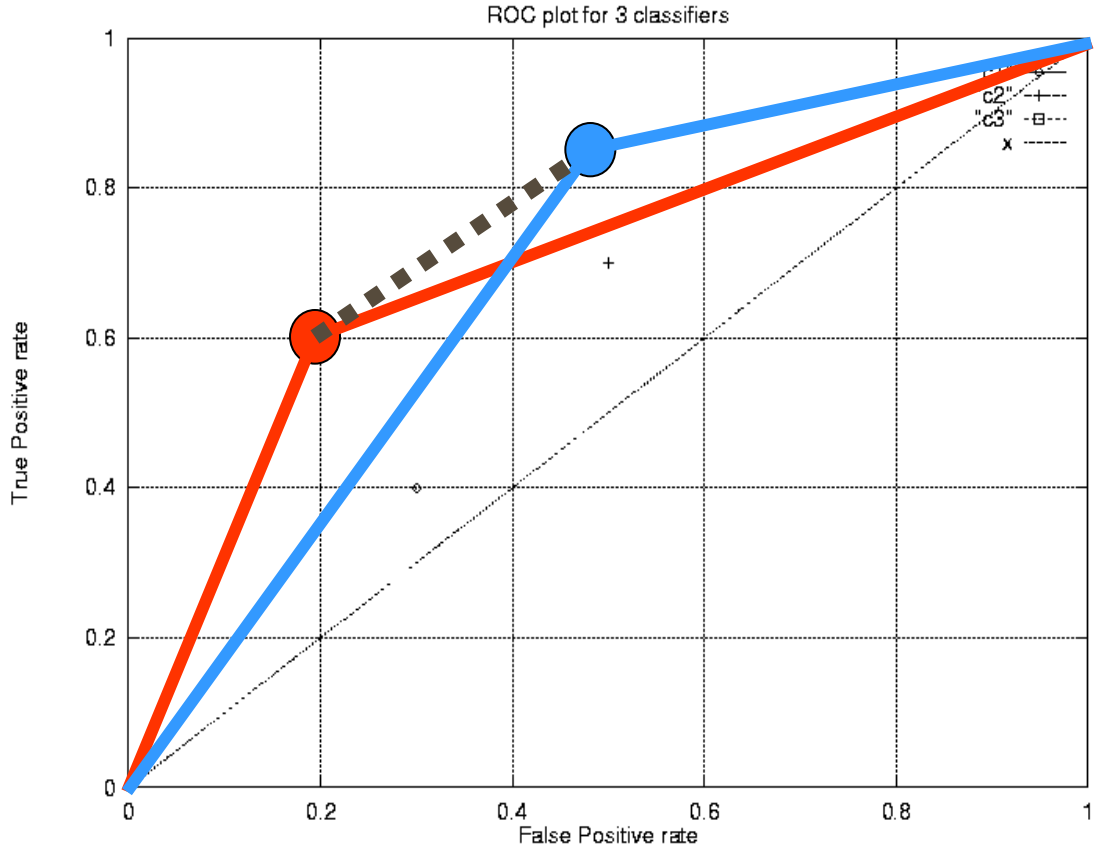
- Supongamos un contexto en el que:
  - Pos = Neg = ½
  - CostFP = 1
  - CostFN = 2
  - Entonces a = (Neg / Pos) * (CostFP / CostFN)
    - = (½ / ½)* ½ = ½
  - y = ½ * x + (1-k/(Pos*CostFN)) = ½ * x +1-k
- Entonces, en este contexto, todos los puntos (x,y) que cumplan y=½*x + 1-0.1 tienen un coste de 0.1
- Para elegir el punto de la curva ROC óptimo para determinadas condiciones operativas, se calculan las líneas de isocoste para esas condiciones y se selecciona el punto ROC correspondiente a la línea de isocoste con menor coste (aquella más cercana a la esquina superior izquierda)

DOMINANCIA EN CURVAS ROC



El clasificador rojo domina al azul y al verde PARA TODOS LOS POSIBLES CONTEXTOS (o condiciones operativas)

CONVEX HULL (ENVOLTURA CONVEXA)



ORGANIZACIÓN

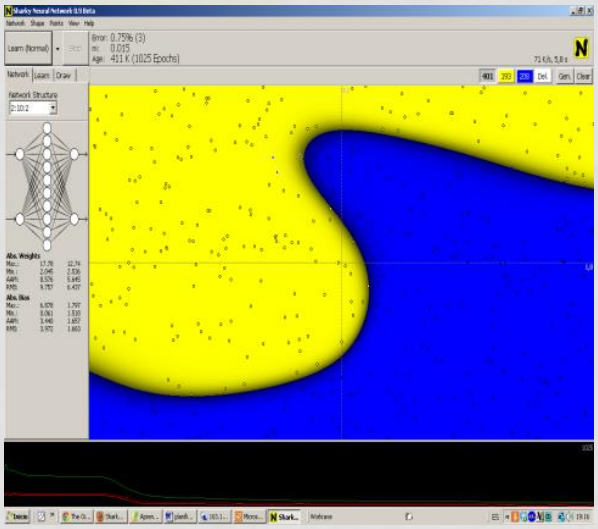
- Evaluación Sensible a la Distribución y al Coste
- Aprendizaje Sensible a la Distribución y al Coste (SMOTE, Metacost)
- Análisis ROC de Clasificadores discretos
- **Análisis ROC de Scorers**
- La Métrica AUC: el área bajo la curva ROC
- Curvas de coste (cost curves)

ANÁLISIS ROC DE SCORERS

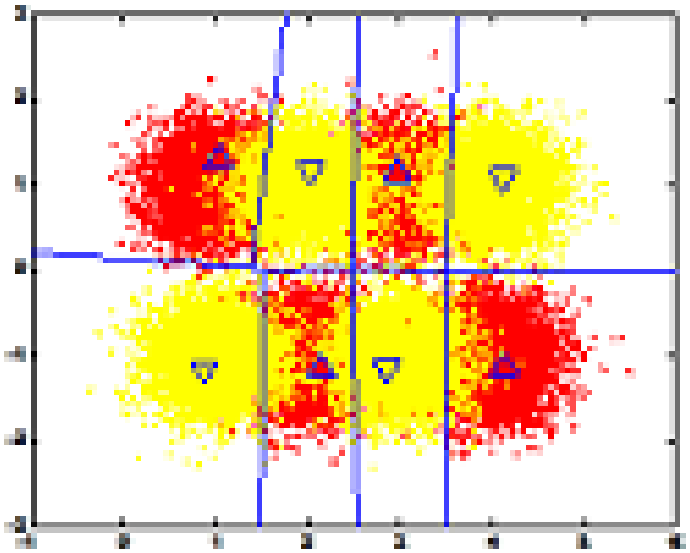
- Clasificadores discretos vs. scorers:
  - Un clasificador discreto predice una clase entre las posibles.
  - Un scorer predice una clase, pero acompaña un valor de fiabilidad a cada predicción.

RECORDATORIO: SCORERS

- ***Clasificación discreta***: $g : X \rightarrow \{0,1\}$
- ***Scoring***: *en caso de que* la función g nos devuelva una medida de pertenencia de una instancia a una clase:
  - $g : X \rightarrow \mathbf{R}$ (en caso de clasificación binaria). Por ejemplo, $g(x)$ = distancia de x a la frontera de separación
    - Si el valor es muy negativo, cercano a la clase 0
    - Si el valor es muy positivo, cercano a la clase 1
    - También podemos normalizar a [0,1]: $g : X \rightarrow [0,1]$

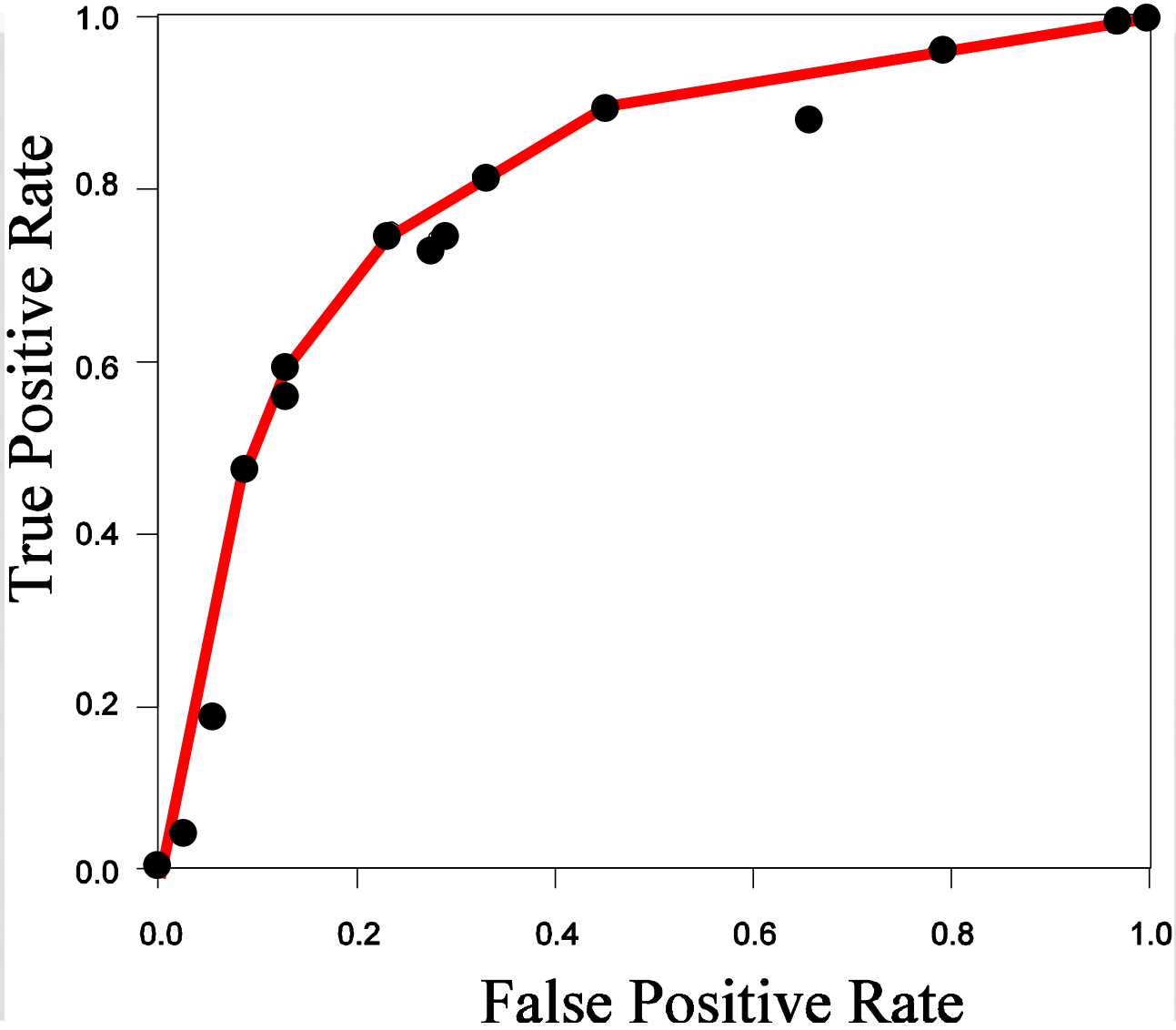


RECORDATORIO: SCORERS,



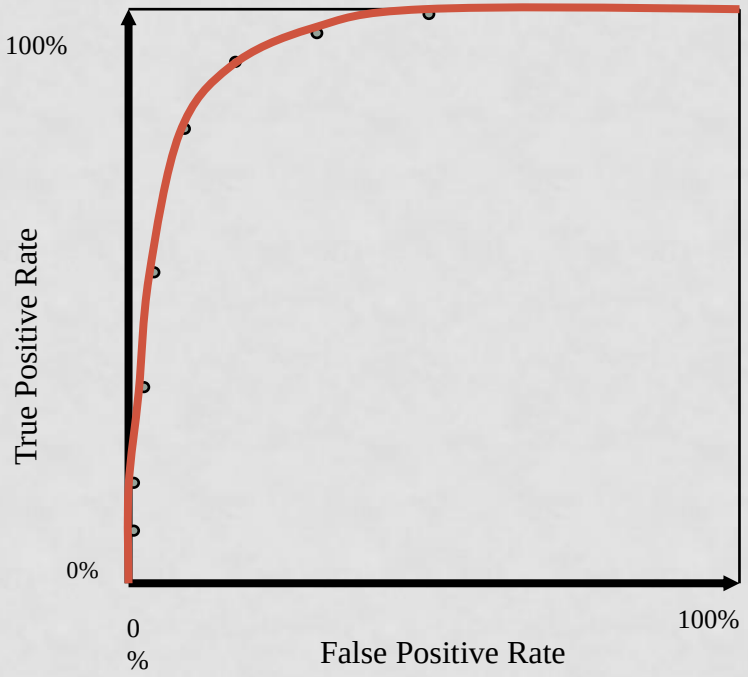
- Es fácil transformar un scorer en un clasificador binario, sin mas que poner un threshold:
  - Si g(x) <= t entonces clase 0
  - Si g(x) > t entonces clase 1
- Eso quiere decir que para distintos t, tenemos distintos clasificadores discretos, es decir, distintos puntos en el espacio ROC

ROC DE UN SCORER PARA DISTINTOS T

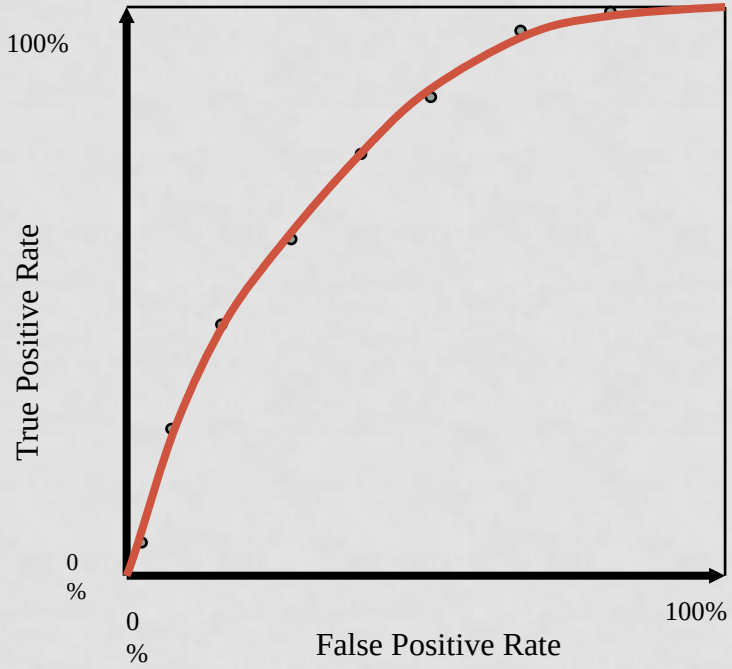


COMPARACIÓN DE CURVAS ROC

Buena



Mala



ORGANIZACIÓN

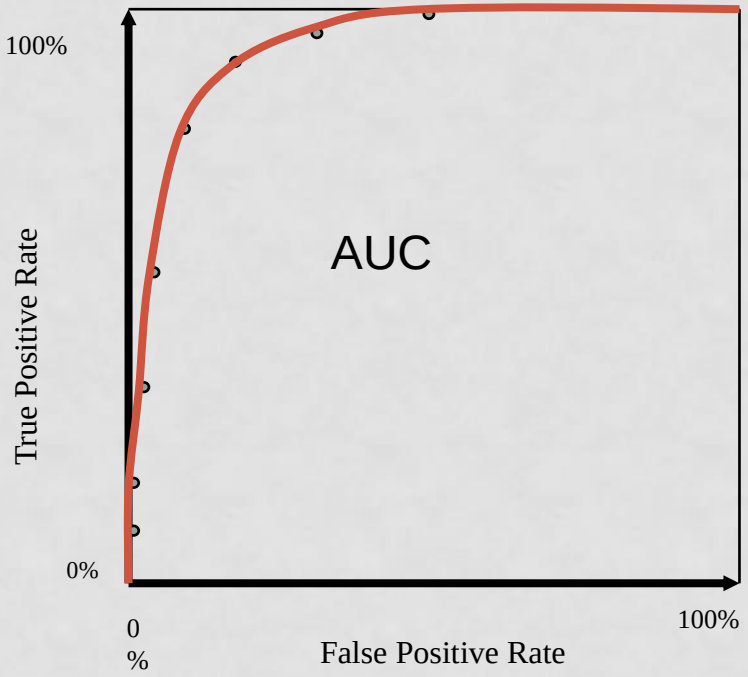
- Evaluación Sensible a la Distribución y al Coste
- Aprendizaje Sensible a la Distribución y al Coste (SMOTE, Metacost)
- Análisis ROC de Clasificadores discretos
- Análisis ROC de Scorers
- **La Métrica AUC: el área bajo la curva ROC**
- Curvas de coste (cost curves)

MÉTRICA AUC: ÁREA BAJO LA CURVA

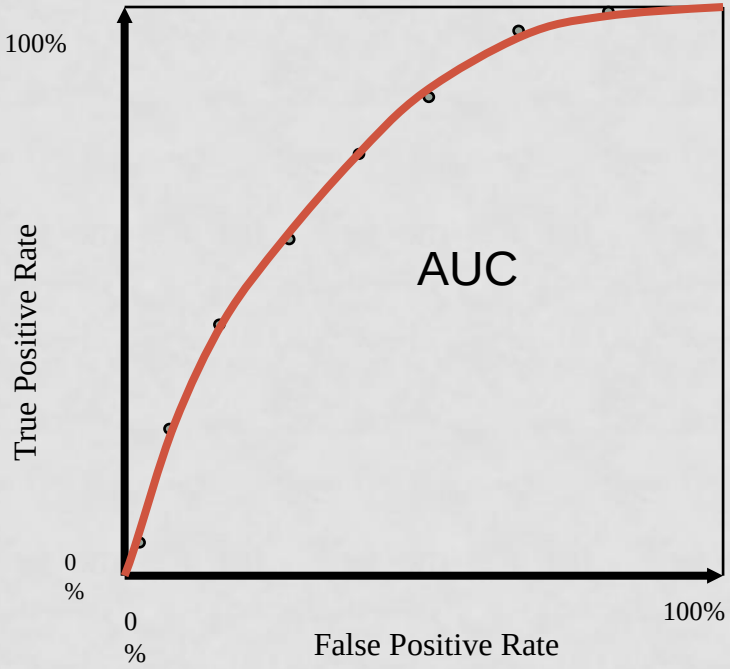
- Se puede utilizar el porcentaje de aciertos para elegir el mejor modelo
- Pero hemos visto que no es muy adecuado si por ejemplo la muestra está desequilibrada
- Existe otra medida que se basa en las curvas ROC y que es inmune al desequilibrio en la muestra: el área bajo la curva de la curva ROC (Area Under the Curve AUC)
- Cuanto mayor sea esa área, más cerca está la curva ROC de la esquina superior izquierda, y por tanto mejor es la separación de los datos (independientemente del desequilibrio de la muestra)

COMPARACIÓN DE CURVAS ROC

Buena



Mala



AUC PARA CLASIFICADORES DISCRETOS

- TP*FN/2 + (1-TP)*(1-FN)/2 + (1-FN)*TP =
- TP*FN/2 + ½ - FN/2 – TP/2 +TP*FN/2 + TP – TP*FN =
- ½ - FN/2 + TP/2 =
- (TN + TP)/2

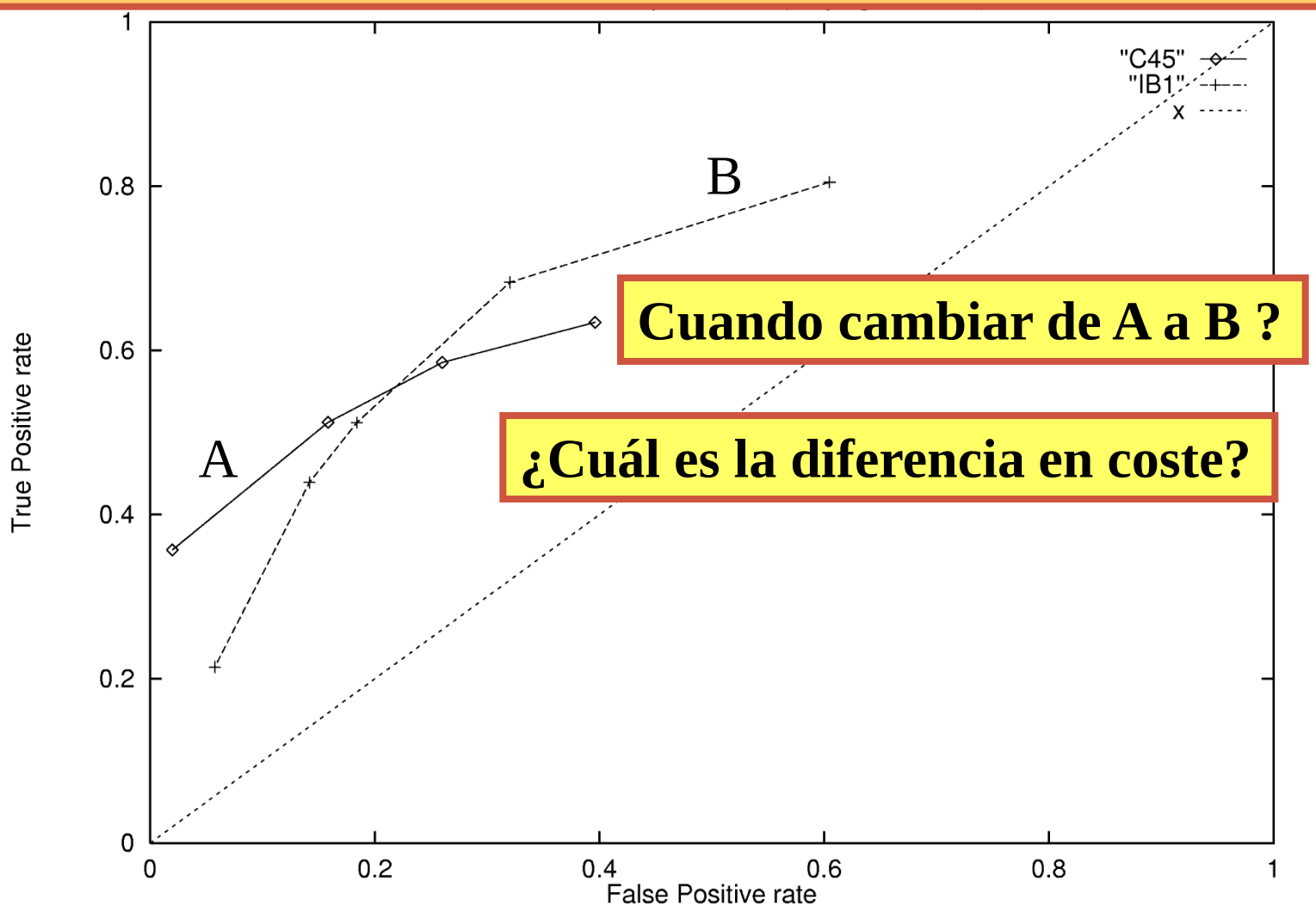
ORGANIZACIÓN

- Evaluación Sensible a la Distribución y al Coste
- Aprendizaje Sensible a la Distribución y al Coste (SMOTE, Metacost)
- Análisis ROC de Clasificadores discretos
- Análisis ROC de Scorers
- La Métrica AUC: el área bajo la curva ROC
- **Curvas de coste (cost curves)**

MOTIVACIÓN

- Las curvas ROC nos permiten elegir el punto operativo óptimo para un contexto determinado (matriz de costes y distribución de datos)
- Sin embargo, no permiten visualizar fácilmente hasta que punto una curva ROC es mejor que otra (en coste)
- Tampoco permiten visualizar fácilmente para que rangos de contextos es mejor una curva que otra
- Fuente (de gráficos): Holte, R. C., & Drummond, C. (2005, August). Cost-sensitive classifier evaluation. In *Proceedings of the 1st international workshop on Utility-based data mining* (pp. 3-9). ACM.

Curvas para dos scorers

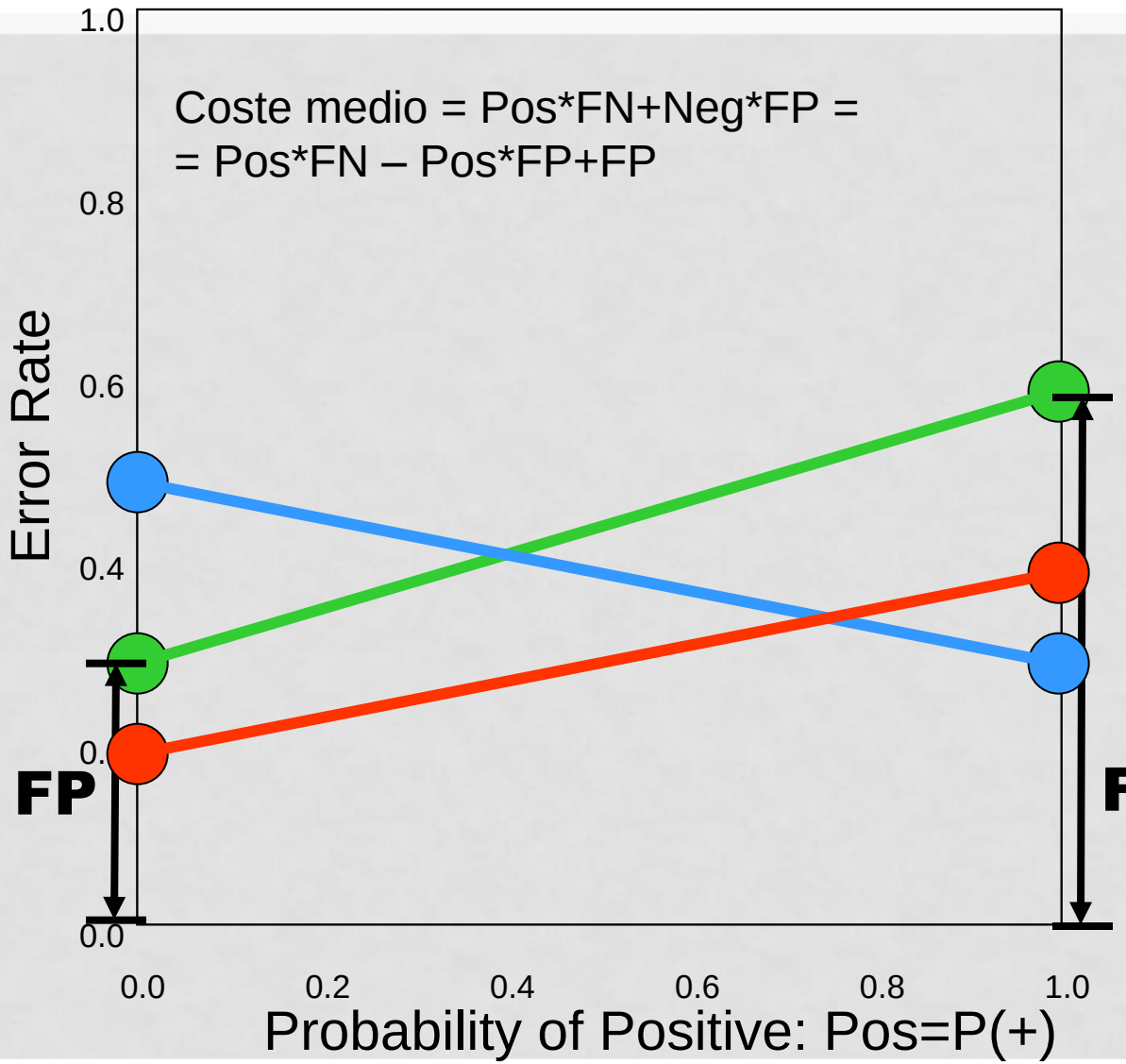


Cuando cambiar de A a B ?

¿Cuál es la diferencia en coste?

- De momento supongamos que no hay costes (o lo que es lo mismo, que todos los costes son 1)
- En ese caso el contexto (que era Pos, Neg y la matriz de costes) queda reducido a Pos (porque Neg = 1 – Pos)
- En este caso, el coste medio es equivalente al error: un fallo cuenta como 1 en ambos casos (confundir + por – o – por +)

COST CURVES



Classifier 1
TP = 0.4
FP = 0.3

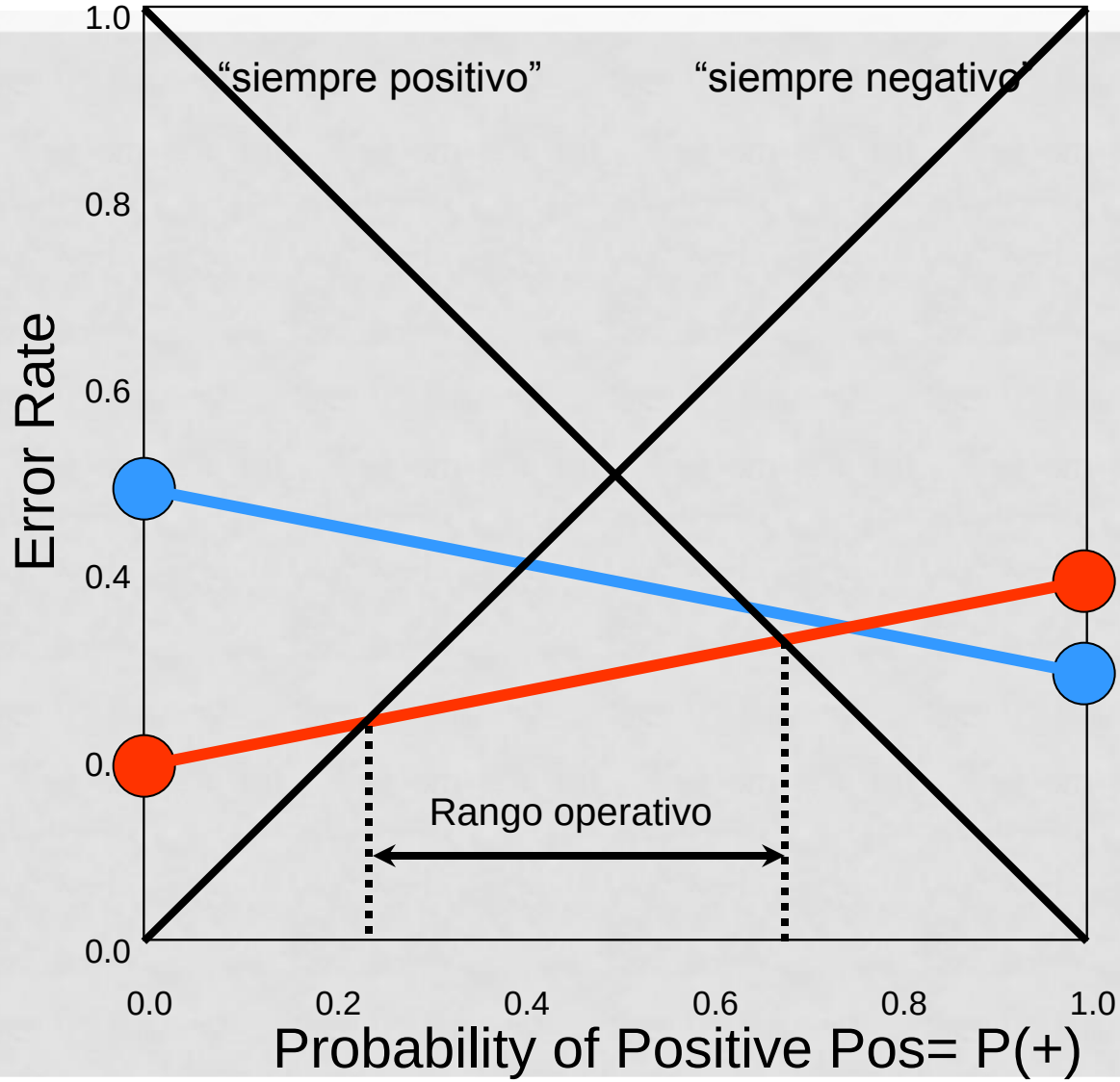
Classifier 2
TP = 0.7
FP = 0.5

Classifier 3
TP = 0.6
FP = 0.2

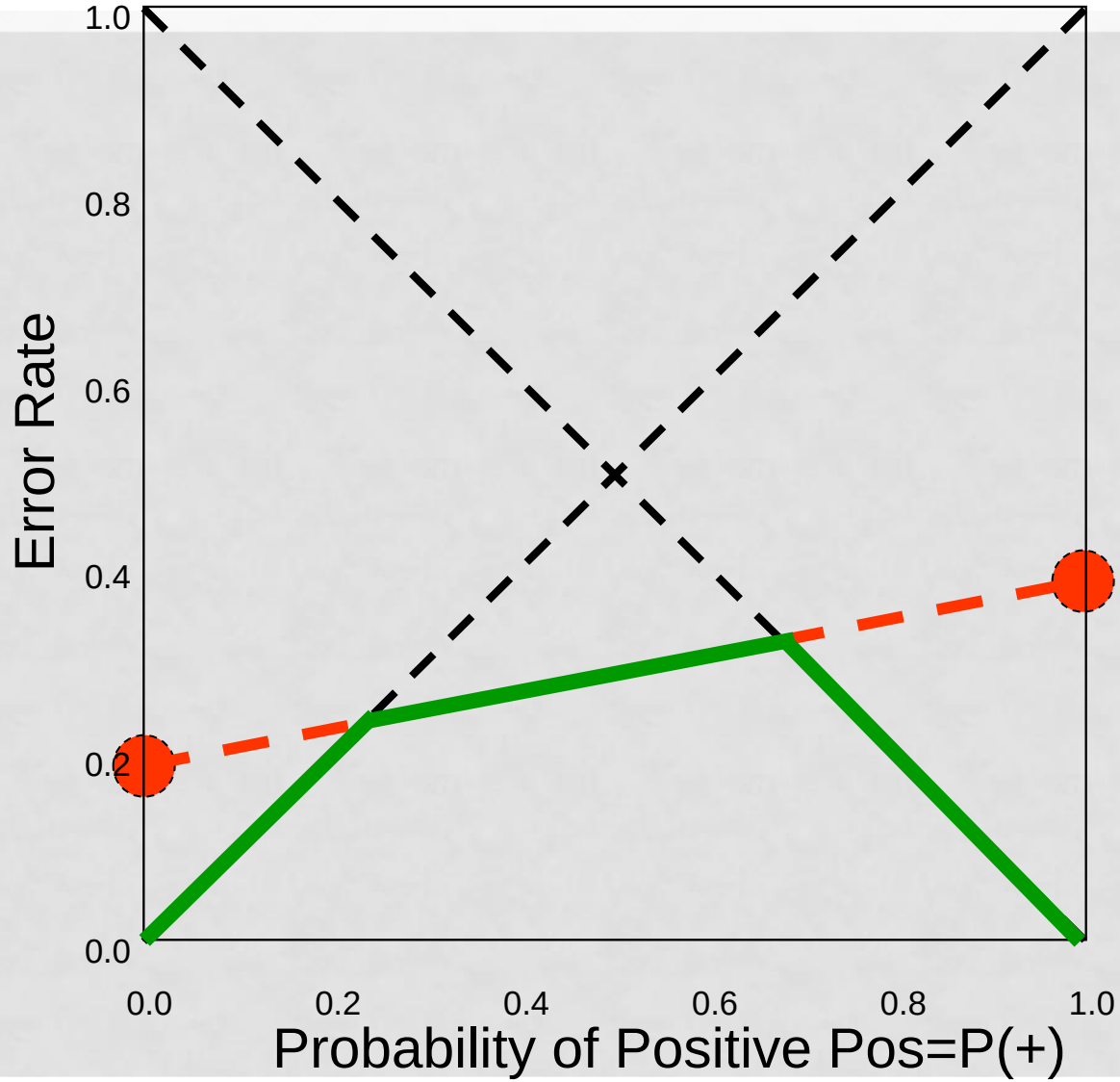
EQUIVALENCIA ROC - CC

- Un punto (FP, TP) en ROC es una línea:
  (0, FP)-(1, FN) en la curva de coste
- Coste medio = Pos*FN*1+Neg*FP*1
  - Pos = p(+)
  - Neg = p(-) = 1-Pos
- Coste medio = Pos*FN – Pos*FP+FP =
  = FP + (FN-FP)* Pos
  - Si Pos = 0 => Coste = FP
  - Si Pos = 1 => Coste = FN

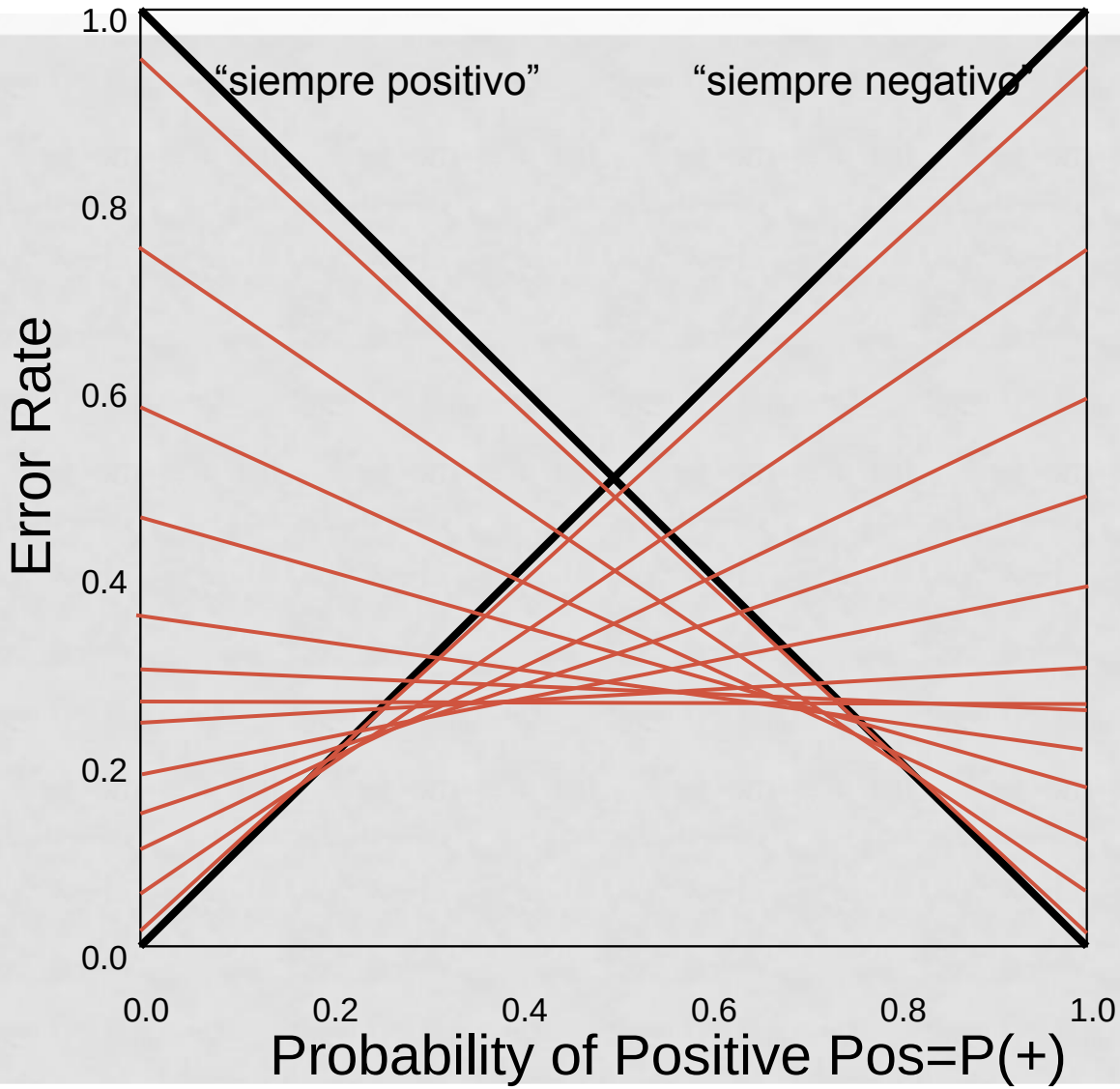
COST CURVES: VISUALIZACIÓN DE RANGOS ÓPTIMOS DEL CONTEXTO



EQUIVALENCIA CON LA ENVOLTURA CONVEXA



COST CURVES


1.0
0.8
0.6
0.4
0.2
0.0
Error Rate
“siempre positivo”
“siempre negativo”
0.0
0.2
0.4
0.6
0.8
1.0
Probability of Positive Pos=P(+)

INCLUYENDO LOS COSTES

- Y=Coste medio = FN*X + FP*(1-X)
- Hasta ahora, X = Pos = p(+)
- Hemos supuesto costes unitarios CostFN = CostFP = 1
- ¿Cómo conseguir curvas de coste con CostFN y CostFP cualesquiera?

INCLUYENDO LOS COSTES

- En lugar de calcular el coste medio, vamos a calcular el coste medio normalizado, para que esté entre cero y uno
- Dividiremos por el coste máximo posible, que ocurre cuando FP = FN = 1
- Sea k(Pos) = coste máximo posible =

  = Pos*FN*CostFN + Neg*FP*CostFP

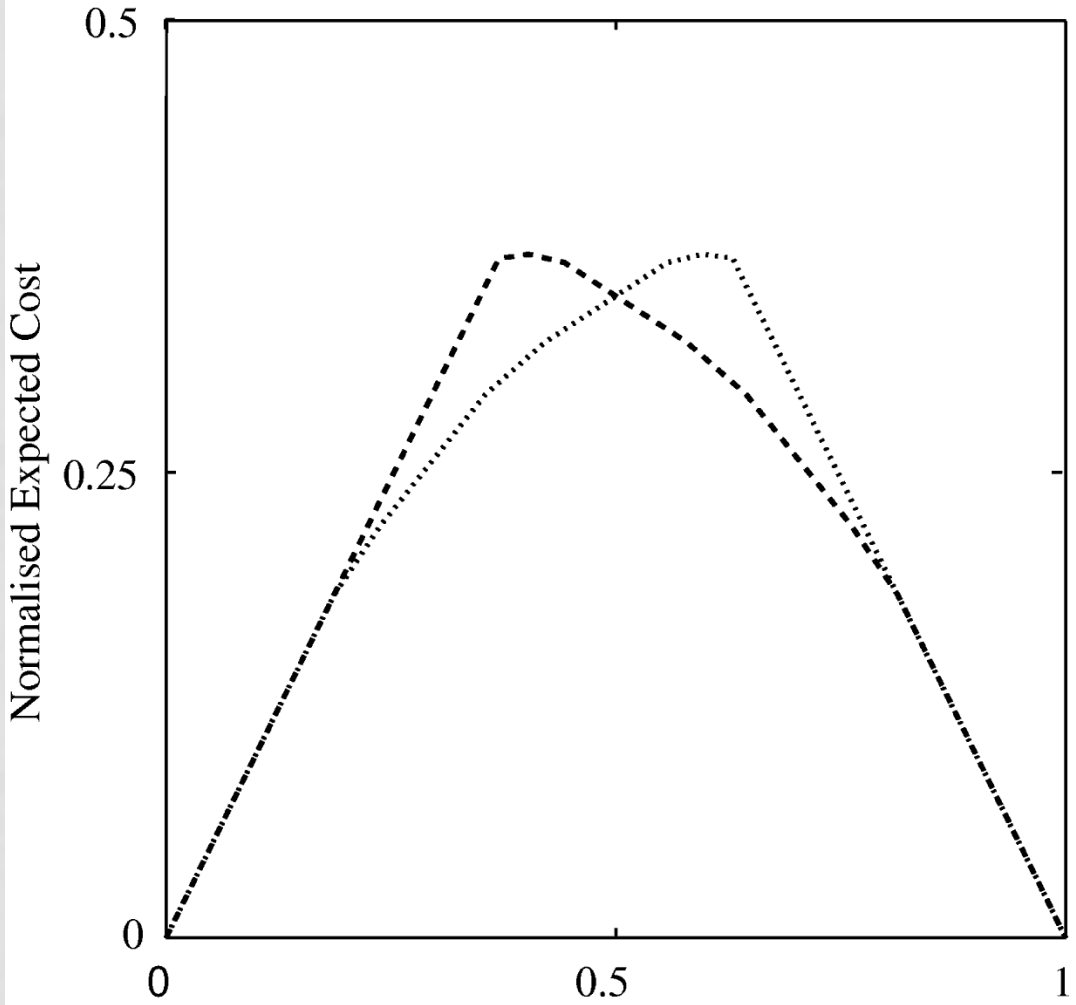
  = Pos*1*CostFN + Neg*1*CostFP

  = Pos*1*CostFN + (1-Pos)*1*CostFP

INCLUYENDO LOS COSTES

- Coste medio normalizado =
  = (Pos*FN*CostFN + Neg*FP*CostFP) / k(Pos)
  = (Pos*CostFN/k(Pos))* FN + (Neg*CostFP/k(Pos))*FP
- Pero tenemos que:
  (Pos*CostFN/k(Pos)) +(Neg*CostFP/k(Pos)) =
  = (Pos*CostFN + Neg*CostFP) / k(Pos) = 1
- Por tanto:
  (Neg*CostFP)/k(Pos) = 1-(Pos*CostFN)/k(Pos)
- Si llamamos X = (Pos*CostFN)/k(Pos)
- Resulta que CMN = Y = X*FN+(1-X)*FP
  - Es como antes solo que ahora X = Pos*CostFN/k(Pos) en lugar de X = Pos

COMPARANDO COST CURVES

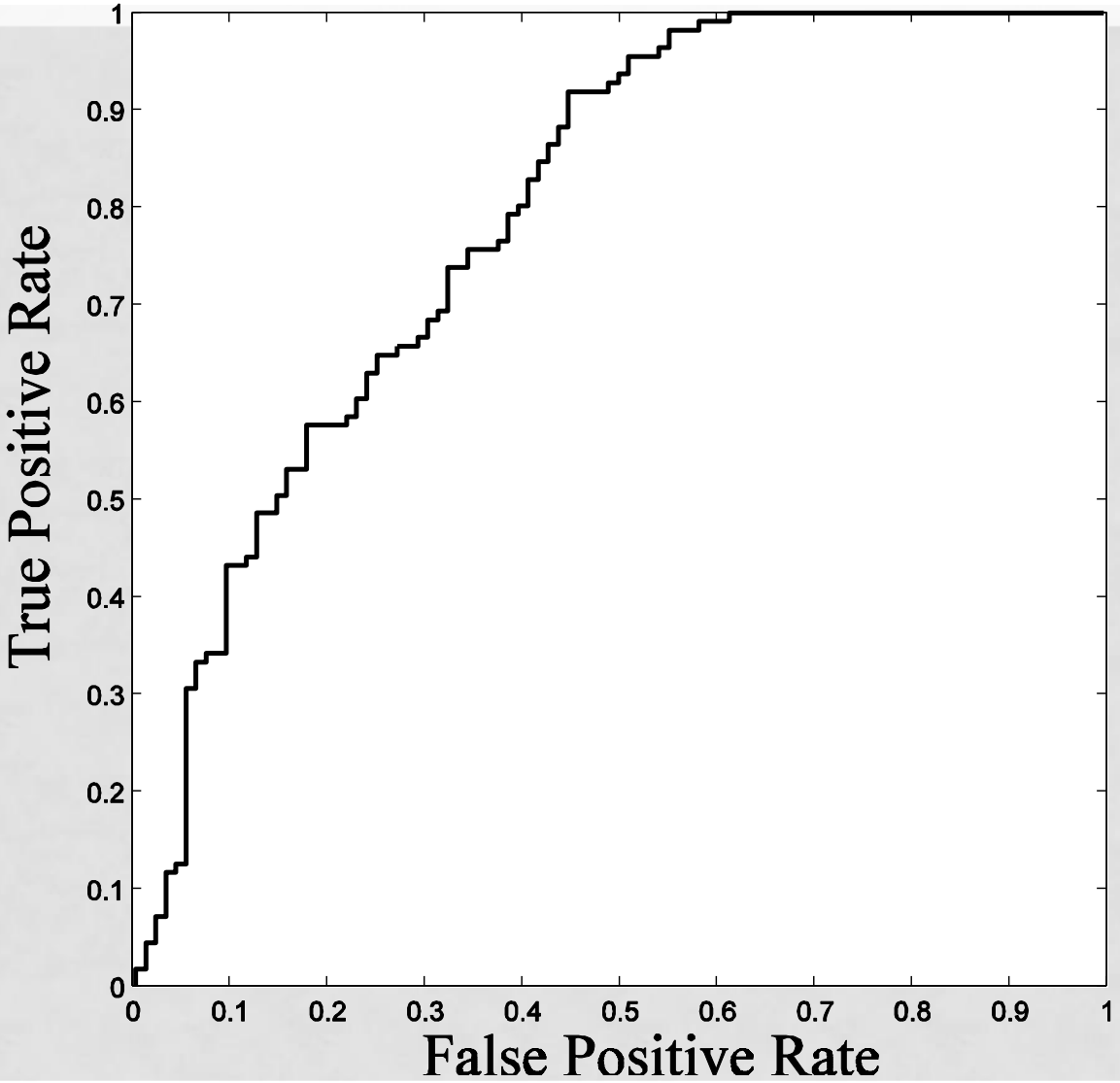


Vemos que para X<0.5 es mejor el primer clasificador

Pos*CostFN/k(Pos) < 0.5

Si conocemos nuestro contexto Pos, CostFN, CostFP, podemos calcular X y ver que scorer tiene un coste menor

ROC, MUCHOS PUNTOS



COST CURVES, MUCHAS LÍNEAS