Introducción a la Cuantificación

Estimando prevalencias por clase mediante aprendizaje supervisado

Juan José del Coz Velasco
juanjo@uniovi.es
Centro de Inteligencia Artificial - Universidad de Oviedo

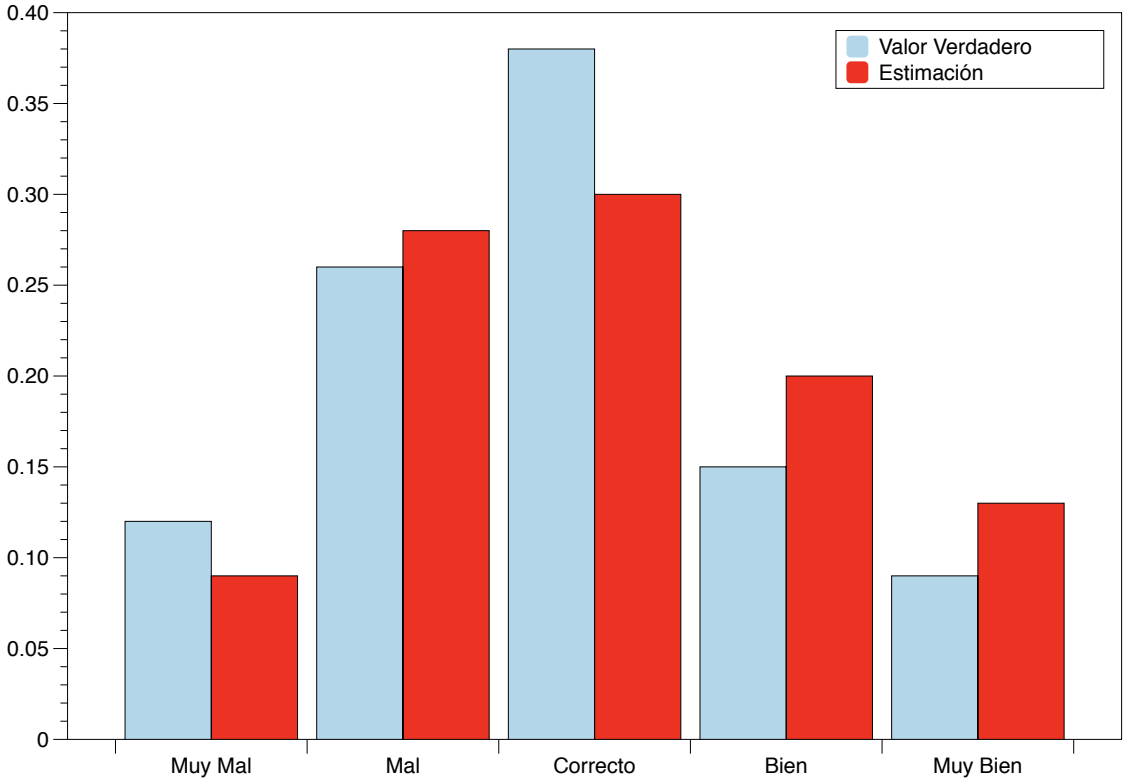
Problema

Problema

Cuantificación

Objetivo

Estimar la distribución de las clases en un conjunto de ejemplos sin etiquetar para el que la predicción de la clase de cada ejemplo es innecesaria



Clasificar y Contar NO funciona

Cuantificación Binaria

Datos: $D^{tr} = \{(x_i^{tr}, y_i^{tr})\}_{i=1}^n \sim S$

$x_i^{tr} \in \mathcal{X}$,

$y_i^{tr} \in \mathcal{Y} = \{-1, +1\}$

Modelo: $\bar{h} : \mathbb{N}^{\mathcal{X}} \longrightarrow [0, 1]$

Predicción: Dado un conjunto, $D^{te} = \{x_j^{te}\}_{j=1}^m \sim T$

$\bar{h}(D^{te}) = \hat{p}$ predice la proporción de positivos en D^{te}
(la proporción de los negativos será $\hat{n} = 1 - \hat{p}$)

Medidas de error:

- de regresión (error absoluto, cuadrático, relativo),
- divergencias (KLD)

Clasificar & Contar (CC)

- Es el método más sencillo
- Entrenamiento: se induce un clasificador binario usando D^{tr}

$$h : \mathcal{X} \longrightarrow \{-1, +1\}$$

- Predicción: se clasifican los ejemplos de D^{te} y se cuentan los que se predicen como positivos

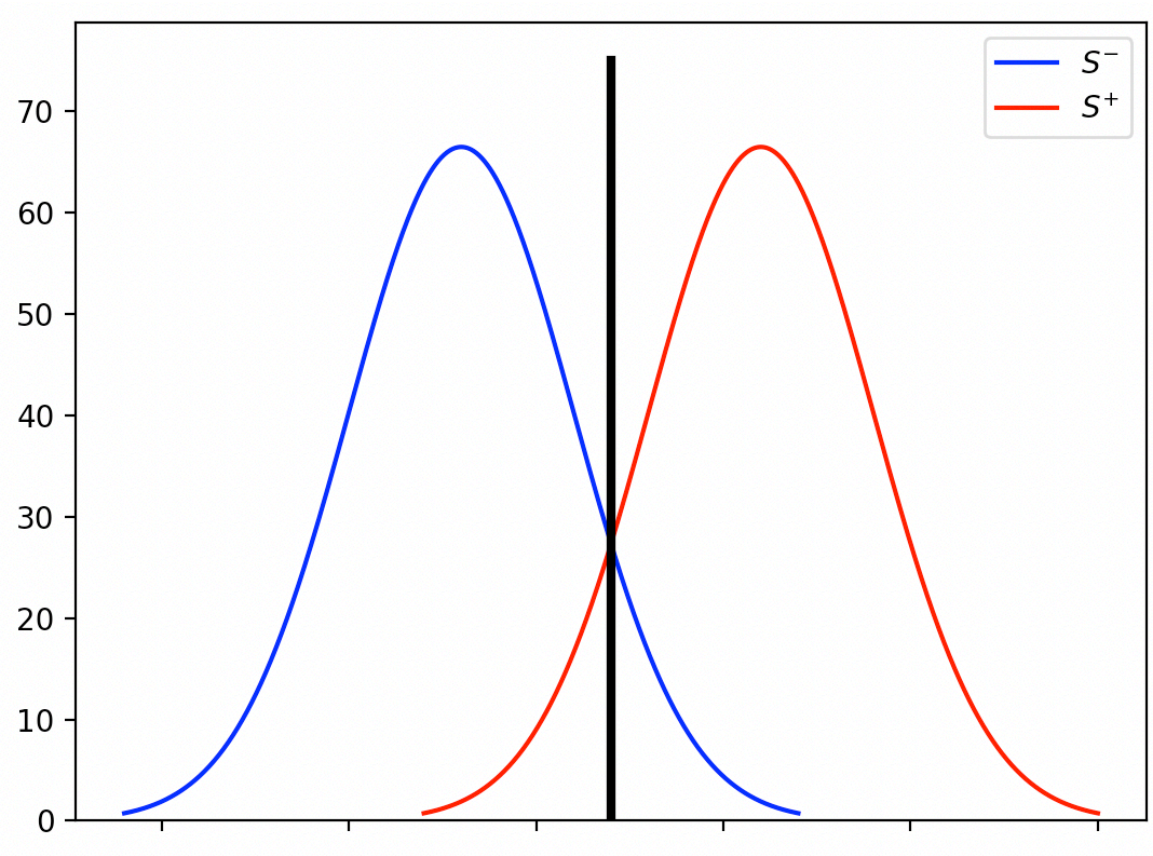
$$\hat{p}_{CC} = \bar{h}(D^{te}) = \frac{1}{m} \sum_{x_j \in D^{te}} I(h(x_j) = +1)$$

Problema

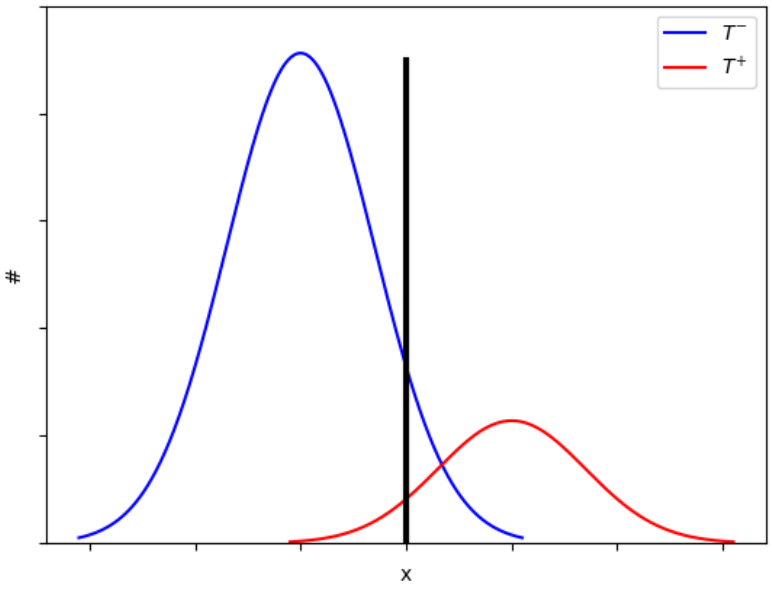
Produce resultados subóptimos:

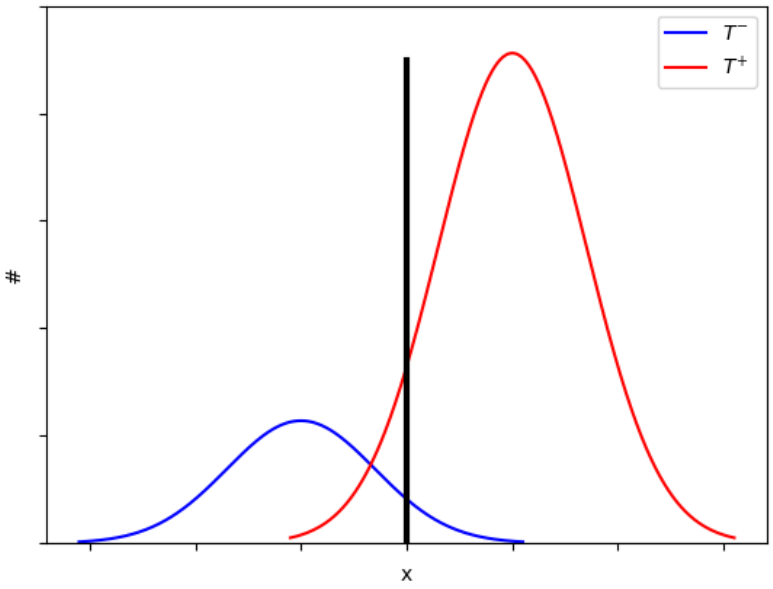
- subestima la prevalencia cuando los positivos crecen, y
- sobreestima la prevalencia cuando los positivos decrecen

Demostración #1: Gráficamente



Demostración #1: Gráficamente





Malas noticias

No funciona porque la distribución de las clases cambia

Demostración #2: Teorema [Forman, 2008]

Relación clave

$$\hat{p}_{CC} \;=\; tpr \cdot p \;+\; fpr \cdot (1-p)$$

Teorema: El método CC, usando un clasificador imperfecto, subestimará p cuando $p > p^*$, y sobreestimará p para $p < p^*$, donde p^* es la proporción en la que el método CC da una estimación correcta.

Demostración: Asumamos que CC retorna una estimación perfecta para un valor p^*. Para un valor $p^* + \epsilon$, $\epsilon \neq 0$, CC no dará la prevalencia perfecta:

$$\begin{aligned} \hat{p}_{CC} &= tpr \cdot (p^* + \epsilon) + fpr \cdot (1 - (p^* + \epsilon)) \\ &= p^* + (tpr - fpr) \cdot \epsilon \end{aligned}$$

- $\hat{p}_{CC} = p^* + \epsilon \iff tpr - fpr = 1$ (clasificador perfecto)
- $tpr - fpr < 1$:
  1. si $\epsilon > 0$, la estimación será más pequeña que $p^* + \epsilon$
  2. if $\epsilon < 0$, la estimación será más grande que $p^* + \epsilon$

Probablistic CC (PCC) [Bella et. al., 2008]

- Entrenamiento: se induce un **clasificador binario probabilístico** usando D^{tr}

$$f : \mathcal{X} \longrightarrow [0,1], \qquad f(x) = P(y = +1|x)$$

- Predicción: se obtiene con f la probabilidad a posteriori de pertenecer a la clase positiva de todos los ejemplos de D^{te} y se calcula la media.

$$\hat{p}_{PCC} = \frac{1}{m} \sum_{x_j \in D^{te}} f(x_j) = \frac{1}{m} \sum_{x_j \in D^{te}} P(y = +1|x_j)$$

Características

- Mismos sesgos que el CC
- Sin embargo, PCC es el cuantificador óptimo en un tipo de problemas (lo discutiremos más adelante)

A. Bella, C. Ferri, J. Hernández-Orallo y M.J. Ramírez-Quintana (2010). "Quantification via Probability Estimators". En: *IEEE ICDM'10*, págs. 737-742

Cambios en la Distribución

Cambios en la Distribución de los Datos

La distribución de los datos cambia por definición del propio problema. Por tanto sabemos que la asunción i.i.d. NO se cumple.

$$P_{tr}(x, y) \neq P_{te}(x, y)$$

Cambios

- $P(y)$ cambia seguro (si no, no habría problema)
- Debemos hacer alguna asunción sobre $P(x)$, $P(y|x)$ y $P(x|y)$ dependiendo de la aplicación a resolver

Problemas y Tipos de Cambios [Moreno et al., 2012]

Problemas:

- $\mathcal{X} \to \mathcal{Y}$: $P(x, y) = P(y|x)P(x)$
- $\mathcal{Y} \to \mathcal{X}$: $P(x, y) = P(x|y)P(y)$

Tipos de Cambios

- ***prior probability shift***: $\mathcal{Y} \to \mathcal{X}$, $P(y)$ cambia, $P(x|y)$ constante
- *covariate shift*: $\mathcal{X} \to \mathcal{Y}$, $P(x)$ cambia, $P(y|x)$ constante
- *concept shift* (o *drift*): $\mathcal{X} \to \mathcal{Y}$, $P(y|x)$ cambia, $P(x)$ constante; $\mathcal{Y} \to \mathcal{X}$, $P(x|y)$ cambia, $P(y)$ constante

J.G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N.V. Chawla y F. Herrera (2012). "A unifying view on dataset shift in classification". En: *Pattern Recognition* 45.1, págs. 521-530

Algoritmos

Adjusted Count (AC) [Forman, 2008]

$$\hat{p}_{CC} \;=\; tpr \cdot p \;+\; fpr \cdot (1-p)$$

Si asumimos que $P(x|y)$ es constante, $P_{tr}(x|y) = P_{te}(x|y)$, entonces tpr y fpr son independientes de los cambios en la distribución y se puede corregir la estimación dada por el método CC:

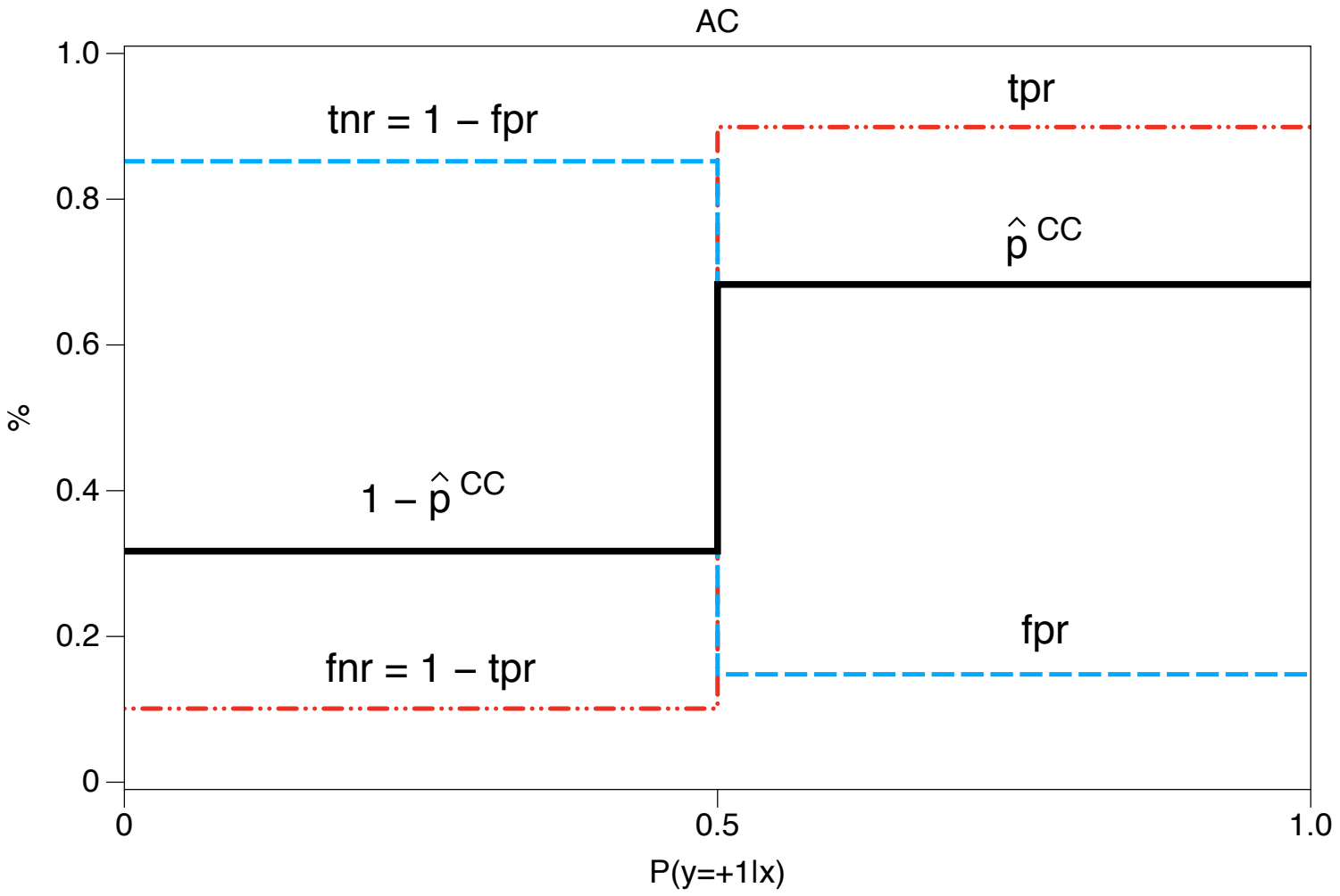
1. Entrenamos un clasificador
2. Estimamos su tpr y fpr (p.e. CV, LOO)
3. Aplicamos el método Classify & Count y obtenemos $\hat{p}_{CC}$
4. Se corrige esa estimación usando:

$$\hat{p}_{AC} = \frac{\hat{p}_{CC} - fpr}{tpr - fpr}$$

5. Nos aseguramos de que el valor esté en $[0, 1]$

G. Forman (2008). "Quantifying counts and costs via classification". En: *Data Mining and Knowledge Discovery* 17.2, págs. 164-206

AC Gráficamente



Probabilistic AC [Bella et. al., 2008]

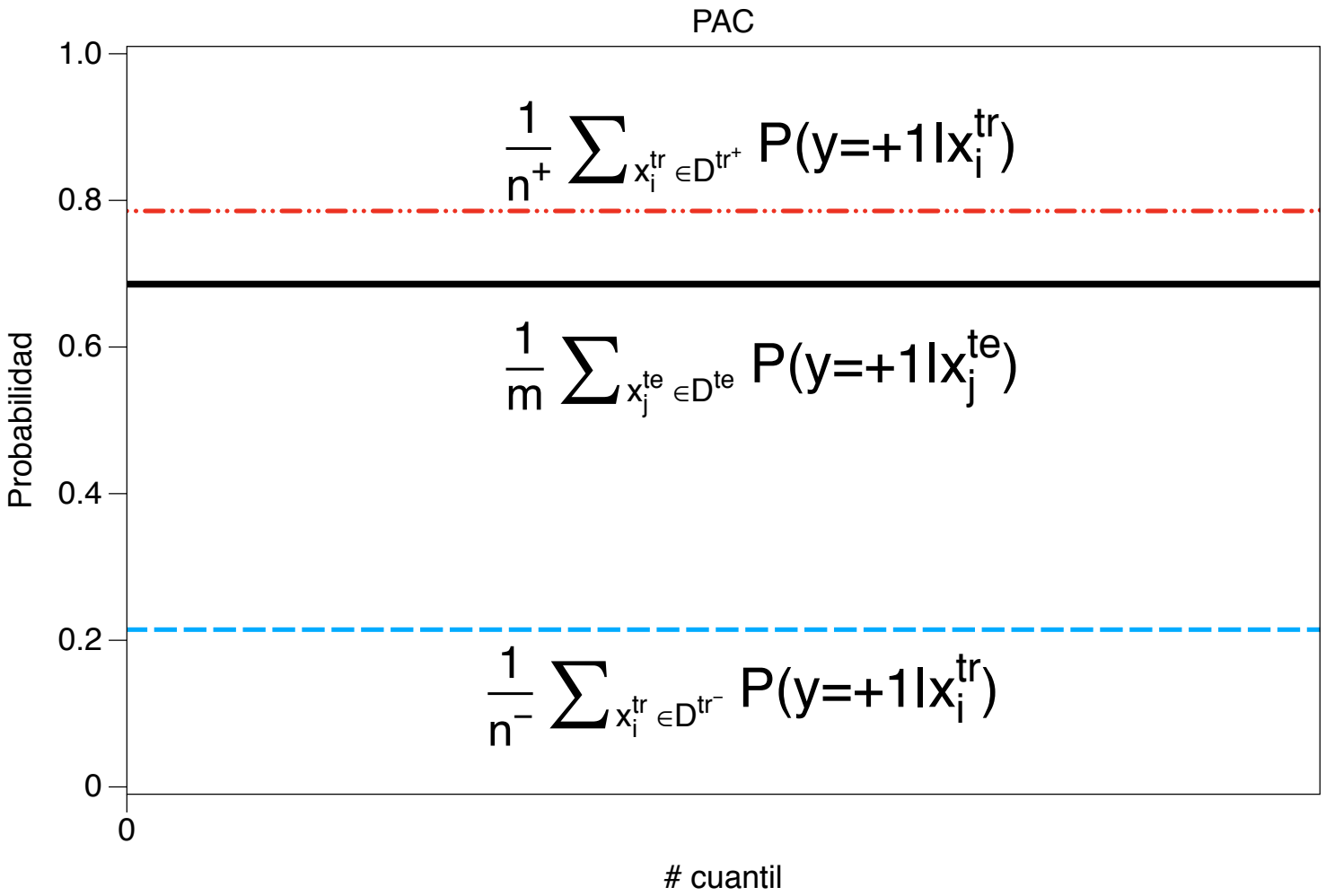
1. Entrena un clasificador binario probabilístico, f
2. Calcula la máxima y la mínima probabilidad media posible:
   - Máxima, todos positivos: $maxPA = \frac{1}{n^+} \sum_{x_i^{tr} \in D^{tr+}} f(x_i^{tr})$
   - Mínima, todos negativos: $minPA = \frac{1}{n^-} \sum_{x_i^{tr} \in D^{tr-}} f(x_i^{tr})$
3. Aplica el clasificador a los ejemplos de test y corrige como el AC:

$$\hat{p}^{PAC} = \frac{\frac{1}{m} \sum_{x_j^{te} \in D^{te}} f(x_j^{te}) - minPA}{maxPA - minPA},$$

4. Nos aseguramos de que el valor esté en $[0, 1]$

A. Bella, C. Ferri, J. Hernández-Orallo y M.J. Ramírez-Quintana (2010). "Quantification via Probability Estimators". En: *IEEE ICDM'10*, págs. 737-742

PAC Gráficamente



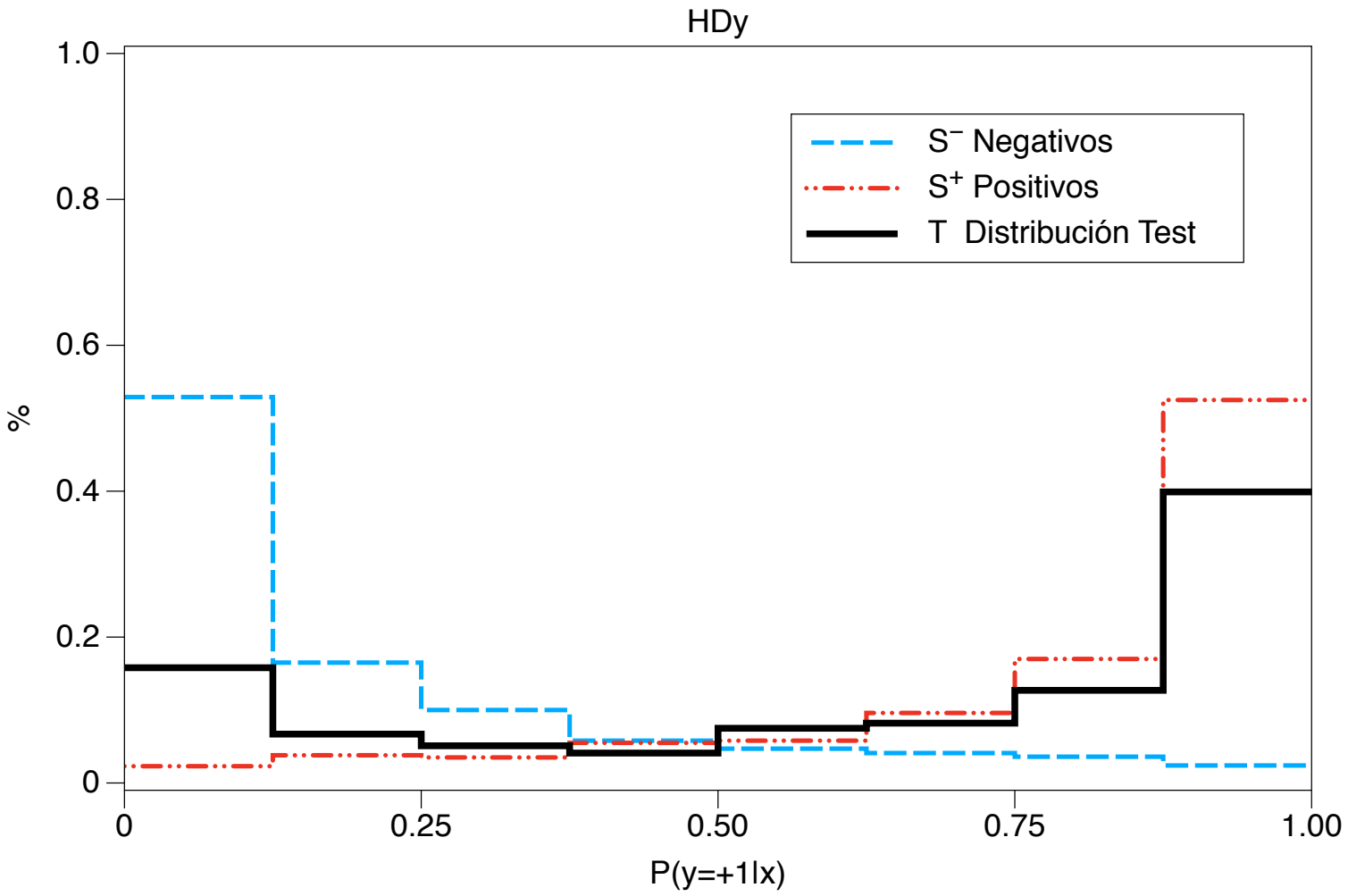
HDy [González-Castro et al, 2012]

1. Entrena un clasificador binario probabilístico, f
2. Calcula histogramas de b particiones de las predicciones de f para la clase positiva y la clase negativa con el conjunto de entrenamiento D^{tr}
3. Calcula el histograma del conjunto de test D^{te}
4. El valor de p que retorna es el valor que minimiza la distancia de Hellinger entre el histograma del conjunto de test y una mezcla ponderada por p de los histogramas de la clase positiva y negativa

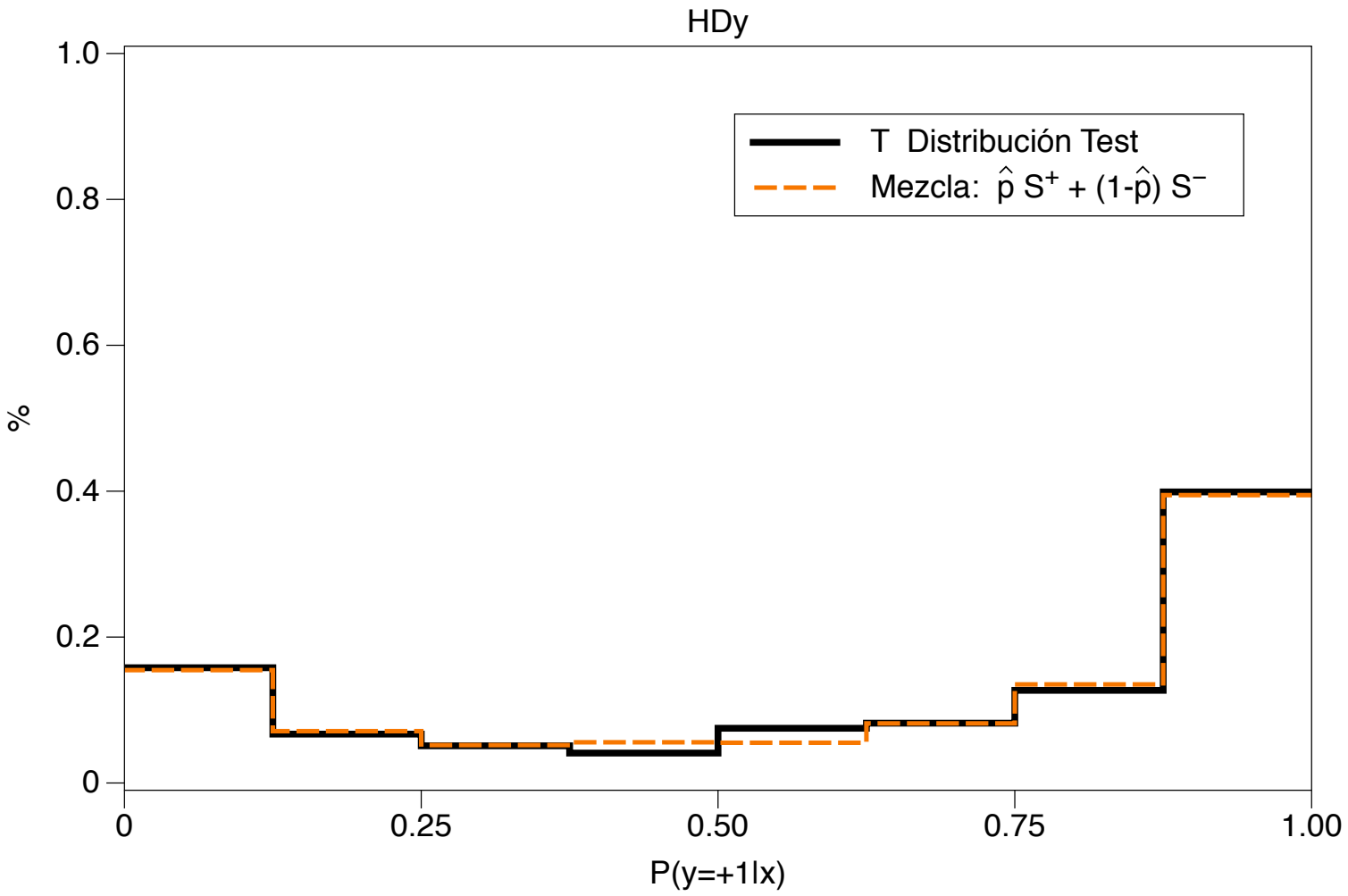
$$\begin{array}{ll} \min\limits_{\hat{p}} & \sqrt{\sum_{k=1}^{b}\left(\sqrt{\frac{|D_{k,l}^{tr-}|}{n^-}(1-\hat{p})+\frac{|D_{k,l}^{tr+}|}{n^+}\hat{p}}-\sqrt{\frac{|D_{k,l}^{te}|}{m}}\right)^2} \\ s.a. & 0 \leq \hat{p} \leq 1. \end{array}$$

Víctor González-Castro, Rocio Alaiz-Rodríguez y Enrique Alegre (2012). "Class Distribution Estimation based on the Hellinger Distance". En: *Information Sciences* 218, págs. 146-164

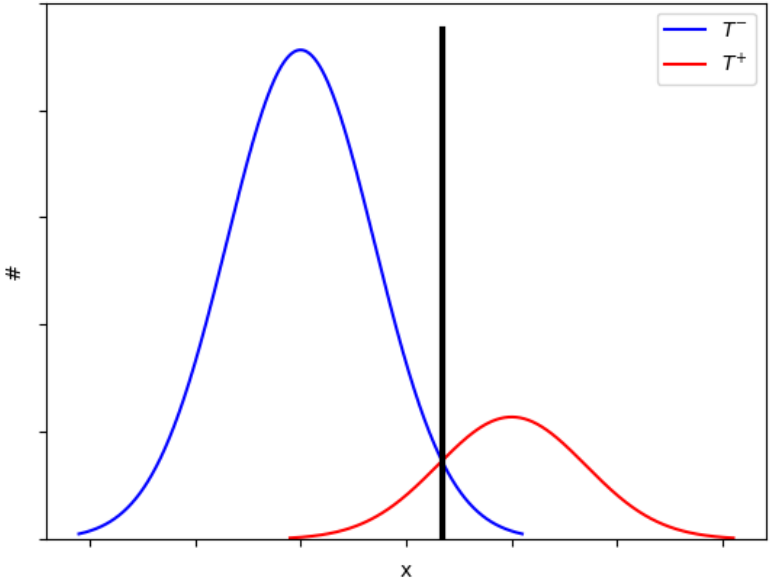
HDy [González-Castro et al, 2012]

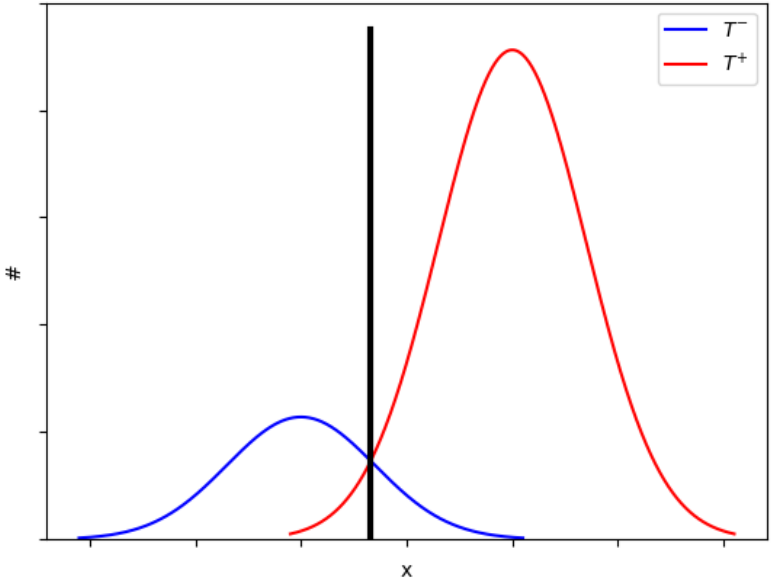


HDy [González-Castro et al, 2012]



Cuantificador basado en EM [Saerens et al 2002]





- Parte de un clasificador probabilístico f y lo "actualiza" para que se ajuste a la distribución de test, pero sin reentrenarlo
- Reajusta las probabilidades a posteriori de todos los ejemplos de D^{te}
- Cuantificar: basta con aplicar PCC con las probabilidades ajustadas

M. Saerens, P. Latinne y C. Decaestecker (2002). "Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure". En: *Neural Computation* 14.1, págs. 21-41

Cuantificador basado en EM [Saerens et al 2002]

Como $P_{tr}(x|y)=P_{te}(x|y)$, aplicando Bayes, $P(A|B)=\frac{P(B|A)\cdot P(A)}{P(B)}$:

$$\frac{P_{tr}(y=+1|x)P_{tr}(x)}{P_{tr}(y=+1)} = \frac{P_{te}(y=+1|x)P_{te}(x)}{P_{te}(y=+1)}$$

Despejando $P_{te}(y=+1|x)$ tenemos:

$$P_{te}(y=+1|x) = \frac{P_{tr}(x)}{P_{te}(x)} \frac{P_{te}(y=+1)}{P_{tr}(y=+1)} P_{tr}(y=+1|x) \tag{1}$$

Calculamos igual $P_{te}(y=-1|x)$ y como $P_{te}(y=+1|x)+P_{te}(y=-1|x)=1$:

$$\frac{P_{tr}(x)}{P_{te}(x)} \left[\frac{P_{te}(y=+1)}{P_{tr}(y=+1)} P_{tr}(y=+1|x) + \frac{P_{te}(y=-1)}{P_{tr}(y=-1)} P_{tr}(y=-1|x) \right] = 1$$

Ajustar probabilidades a posteriori

$$P_{te}(y=+1|x) = \frac{\frac{P_{te}(y=+1)}{P_{tr}(y=+1)} P_{tr}(y=+1|x)}{\frac{P_{te}(y=+1)}{P_{tr}(y=+1)} P_{tr}(y=+1|x) + \frac{P_{te}(y=-1)}{P_{tr}(y=-1)} P_{tr}(y=-1|x)}$$

Cuantificador basado en EM [Saerens et al 2002]

Dado un conjunto $D^{te} = \{x_1, \ldots, x_m\}$ maximizamos la verosimilitud:

$$L(x_1, \ldots, x_m) = \prod_{i=1}^{m} P_{te}(x_i|y=+1)P_{te}(y=+1) + P_{te}(x_i|y=-1)P_{te}(y=-1)$$

Algoritmo: modelo $P_{te}(y=+1)$, vbles ocultas $P_{te}(y=+1|x_i)\ \forall x_i \in D^{te}$

$$\text{Inicio}: P_{te}^{0}(y=+1) = P_{tr}(y=+1)$$

$$\text{Paso E}: P_{te}^{k}(y=+1|x_i) = \frac{\frac{P_{te}^{k}(y=+1)}{P_{tr}(y=+1)} P_{tr}(y=+1|x_i)}{\frac{P_{te}^{k}(y=+1)}{P_{tr}(y=+1)} P_{tr}(y=+1|x_i) + \frac{P_{te}^{k}(y=-1)}{P_{tr}(y=-1)} P_{tr}(y=-1|x_i)}$$

$$\text{Paso M}: P_{te}^{k+1}(y=+1) = \frac{1}{m} \sum_{i=1}^{m} P_{te}^{k}(y=+1|x_i)$$

Stop: nº máximo de iteraciones o $|P_{te}^{k+1}(y=+1) - P_{te}^{k}(y=+1)| < \epsilon$

Tipos de Algoritmos

Dos Tipos de Algoritmos de Cuantificación

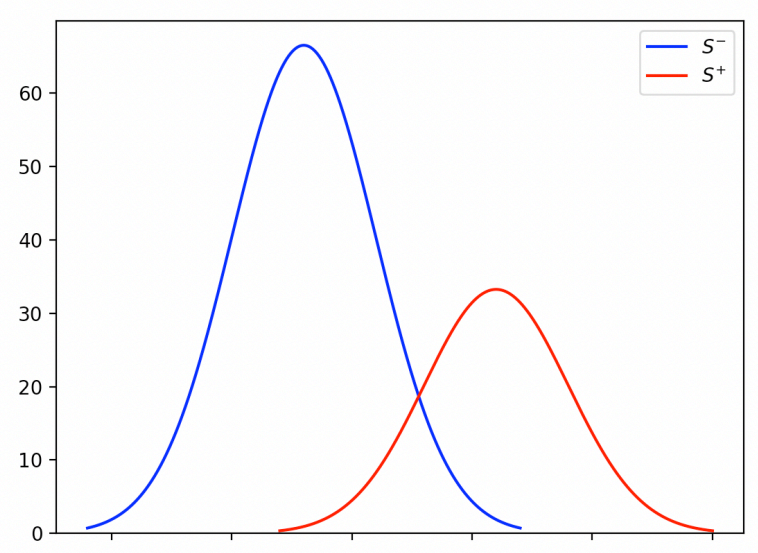
El problema de cuantificación, bajo la asunción de prior probability shift, se ha resuelto usando dos técnicas de aprendizaje conocidas:

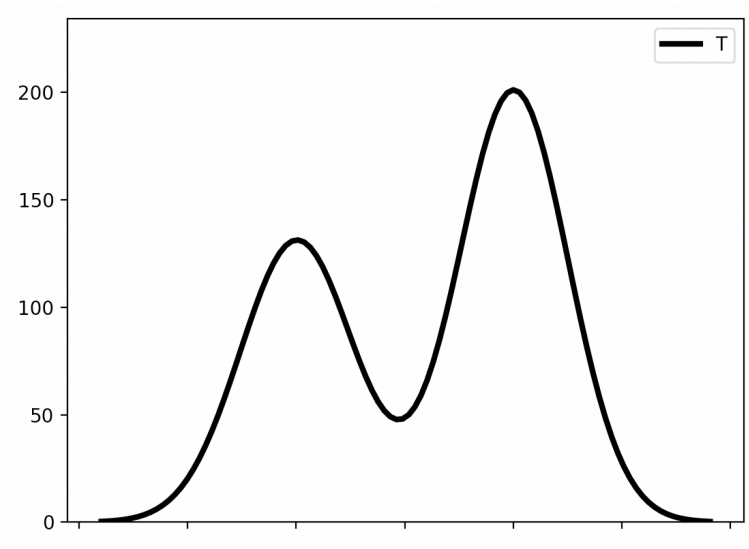
- **Estimar densidades**: si eres capaz de estimar bien las distribuciones, solo tienes que mezclarlas y ajustarlas para obtener la prevalencia real. Ejemplos: todos los algoritmos que hemos visto salvo el EM.
- **Calibración**: si eres capaz de obtener un clasificador probabilístico bien calibrado con los datos de entrenamiento, entonces tienes resuelto el problema de cuantificación. Ejemplo: método EM.

Consistencia de Fisher

Ambos enfoques cumplen con la consistencia de Fisher. Significa que el error en cuantificación disminuye a medida que aumenta el tamaño de la muestra de test

Estimar y Ajustar Distribuciones





$$\begin{aligned} \min_{\hat{p}} \quad & \Delta(T, S^- \cdot (1-\hat{p}) + S^+ \cdot \hat{p}), \\ s.a. \quad & 0 \leq \hat{p} \leq 1. \end{aligned}$$

- Esta familia de métodos varía en tres aspectos:
  1. la métrica Δ para medir la similitud entre las distribuciones,
  2. la forma de representar las distribuciones, y
  3. cómo buscar el valor óptimo para $\hat{p}$.
- Se asume $P(x|y)$ cte, S^+ y S^- varían uniformemente con $\hat{p}$

Estimar y Ajustar Distribuciones

Usan normalmente las predicciones de un clasificador y no el espacio original de atributos

- Es más conveniente porque estimas la densidad en un espacio más pequeño, en concreto c-1 dimensiones siendo c el número de clases
- Produce mejores resultados
- Y es correcto, cumple la asunción de prior probability shift

Lema [Lipton et al. 2018]: Si $P_{tr}(x|y) = P_{te}(x|y)$ es cierto, entonces también es cierto que $P_{tr}(f(x)|y) = P_{te}(f(x)|y)$, siendo f un clasificador.

Zachary C Lipton, Yu-Xiang Wang y Alex Smola (2018). "Detecting and Correcting for Label Shift with Black Box Predictors". En: *ICML 2018*

Estimar y Ajustar Distribuciones: Métodos

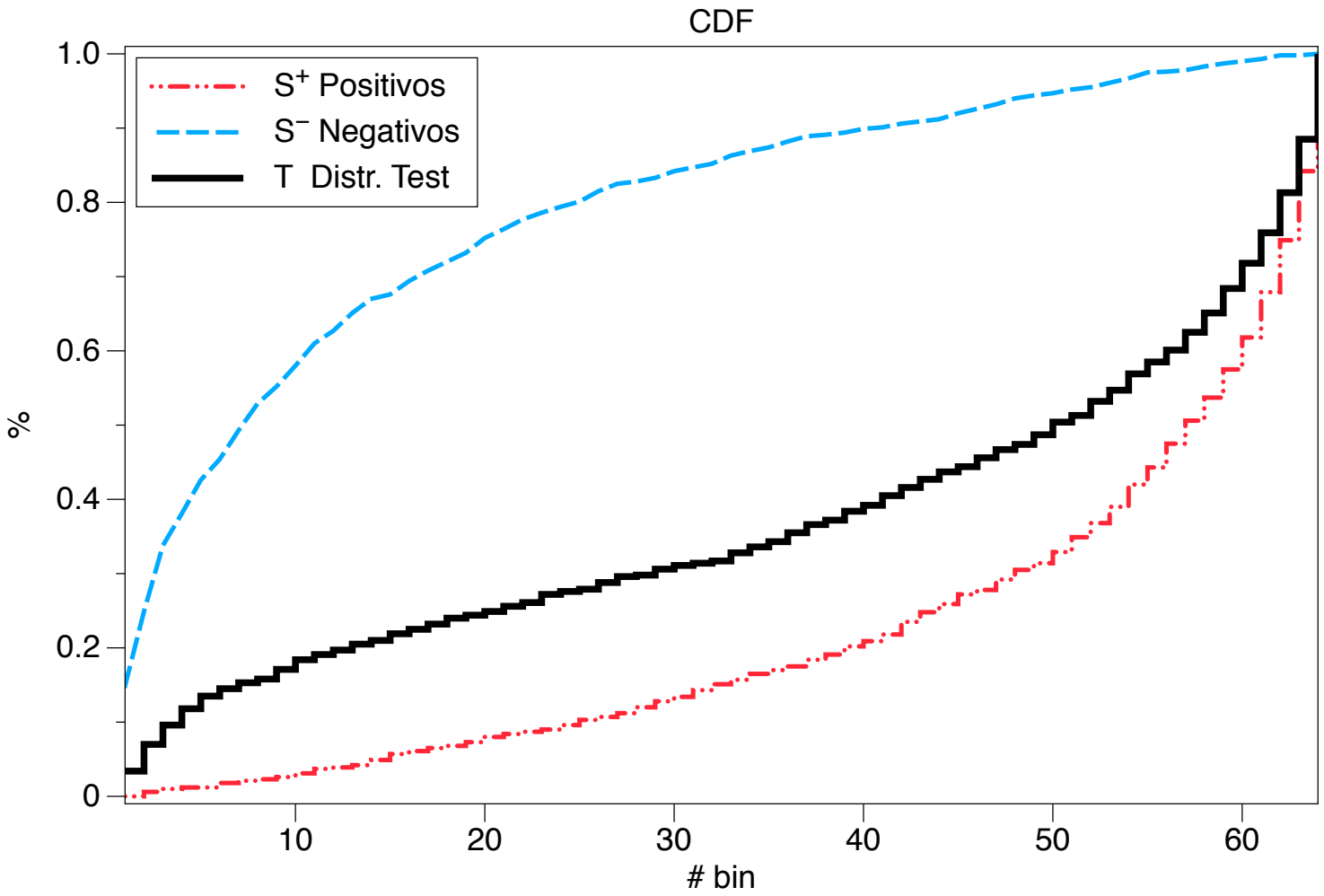
Métodos para estimar las densidades:

- PDFs (histogramas): AC [Forman 2008], HDX, HDy [González-Castro et al, 2012]
- CDFs: MM [Forman 2008], ORD [Maletzke et al. 2019]
- Cuantiles: PAC [Bella et al. 2010], QUANTy [Castaño et al. 2022]
- Conjuntos: EDX [Kawakubo et al. 2016], EDy [Castaño et al. 2018]

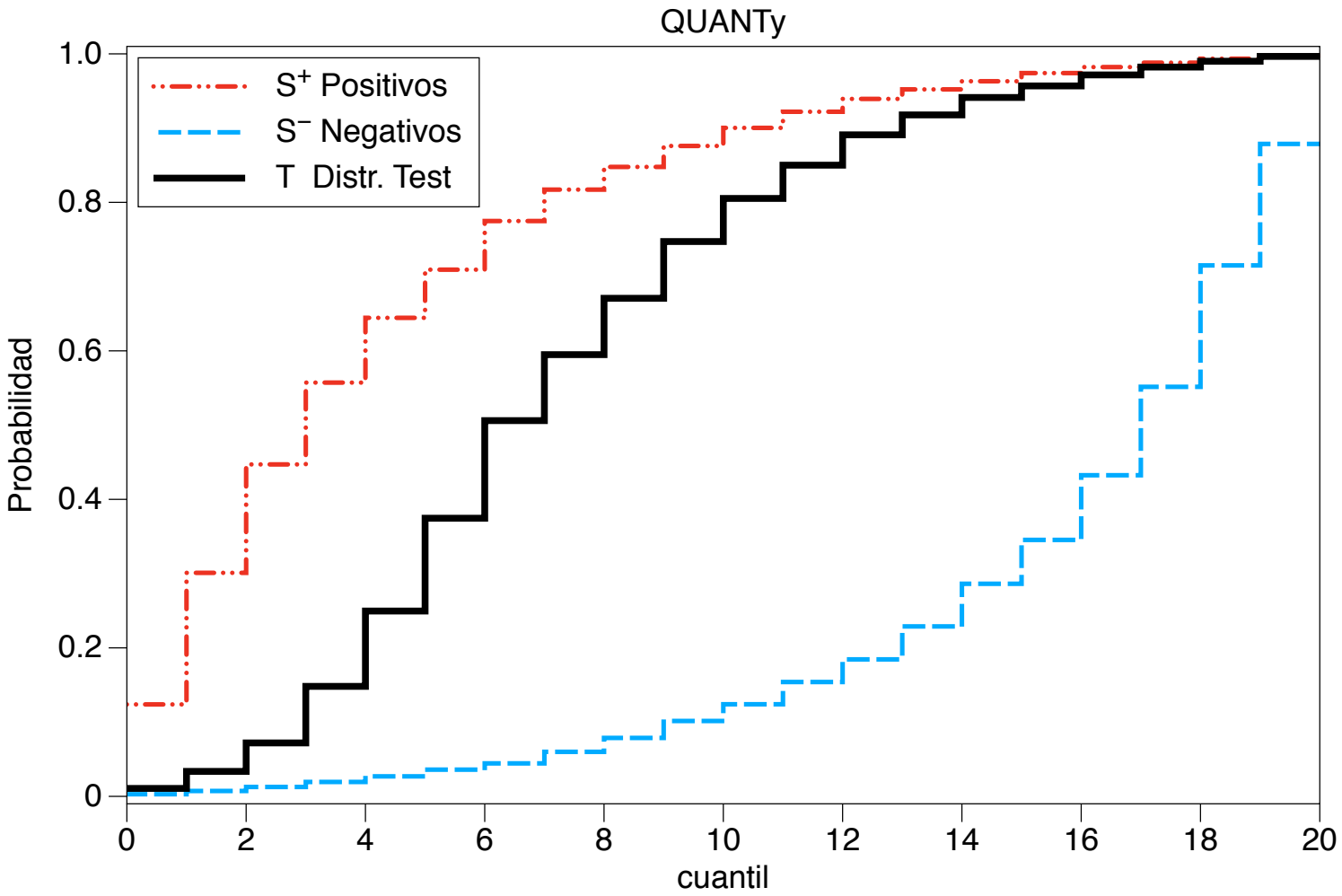
Medidas de similitud entre distribuciones (Δ):

- Hellinger Distance: HDX, HDy [González-Castro et al, 2012]
- Energy Distance: EDx [Kawakubo et al. 2016], EDy [Castaño et al. 2018]
- Earth Mover's Distance (EMD): [Maletzke et al. 2019]
- Normas: L1 [Forman 2008], L2 [Firat 2016]

CDFs


CDF
1.0
0.8
0.6
0.4
0.2
0
%
10
20
30
40
50
60
bin
S+ Positivos
S− Negativos
T Distr. Test

Cuantiles


QUANTy
S+ Positivos
S− Negativos
T Distr. Test
Probabilidad
cuantil
0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
12
14
16
18
20

Calibración del Clasificador

- Puedes aplicar cualquier método de calibración existente
- Depende a veces del clasificador que utilices
- El método basado EM [Saerens et al. 2002] es uno de los mejores algoritmos de cuantificación
- Pero es altamente dependiente de la calibración
- Los métodos basados en ajustes de distribución son más estables, aunque a veces puedan estimar peor que el método basado en EM

Amr Alexandari, Anshul Kundaje y Avanti Shrikumar (2020). "Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation". En: *International Conference on Machine Learning*. PMLR, págs. 222-232

¿Qué pasa si no hay prior probability shift?

Covariate shift, $\mathcal{X} \to \mathcal{Y}$, $P(x)$ cambia, $P(y|x)$ constante

- **Probabilistic CC es óptimo para covariate shift** ya que estima $P(y|x)$ y eso no cambia aunque cambie $P(x)$

Concept shift, cambia el concepto de la clase, dos casos:

- $\mathcal{X} \to \mathcal{Y}$, $P(y|x)$ cambia, $P(x)$ constante: estimar las prevalencias a partir de datos etiquetados pasaría por aprender el nuevo concepto
- $\mathcal{Y} \to \mathcal{X}$, $P(x|y)$ cambia, $P(y)$ constante: no es un problema de cuantificación ya que $P(y)$ es constante

Cuantificación Multiclase

Cuantificación Multiclase

Datos: $D^{tr} = \{(x_i^{tr}, y_i^{tr})\}_{i=1}^n \sim S$

$x_i^{tr} \in \mathcal{X}$,

$y_i^{tr} \in \mathcal{Y} = \{1, 2, \ldots, k\}$

Modelo: $\bar{h} : \mathbb{N}^{\mathcal{X}} \longrightarrow [0,1]^k$

Predicción: Dado un conjunto, $D^{te} = \{x_j^{te}\}_{j=1}^m \sim T$

$$\bar{h}(D^{te}) = [\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_k], \quad \hat{p}_j \geq 0, \sum_{j=1}^{k} \hat{p}_j = 1$$

Medidas de error:

- de regresión multiobjetivo (error absoluto, cuadrático, relativo),
- divergencias (KLD)

Algoritmos de Cuantificación Multiclase

Los cuantificadores que hemos visto son extensibles fácilmente a multiclase

AC: $P_{D^{te}}(h(x) = c_a) = \sum_{l=1}^{k} P(h(x) = c_a | y = c_l) P_{D^{te}}(y = c_l)$

$$\begin{pmatrix} P_{D^{tr}}(c_1|c_1) & P_{D^{tr}}(c_1|c_2) & P_{D^{tr}}(c_1|c_3) \\ P_{D^{tr}}(c_2|c_1) & P_{D^{tr}}(c_2|c_2) & P_{D^{tr}}(c_2|c_3) \\ P_{D^{tr}}(c_3|c_1) & P_{D^{tr}}(c_3|c_2) & P_{D^{tr}}(c_3|c_3) \end{pmatrix} * \begin{pmatrix} \hat{p}_1 \\ \hat{p}_2 \\ \hat{p}_3 \end{pmatrix} = \begin{pmatrix} P_{D^{te}}(h(x)=c_1) \\ P_{D^{te}}(h(x)=c_2) \\ P_{D^{te}}(h(x)=c_3) \end{pmatrix}$$

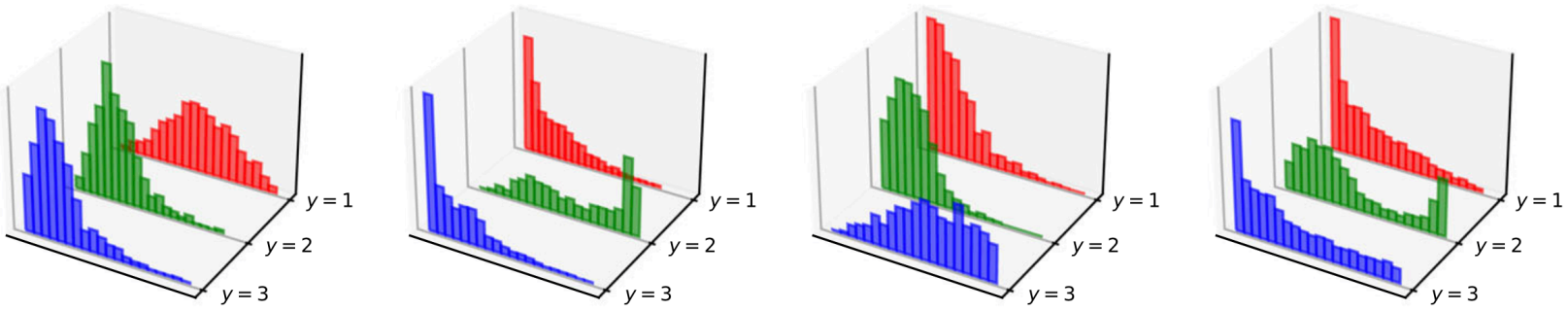
PAC: Sistema de ecuaciones similar a AC pero con medias de probabilidades

EM:

$$P_{D^{te}}(y = c_a | x_i) = \frac{\frac{P_{D^{te}}(y=c_a)}{P_{D^{tr}}(y=c_a)} P_{tr}(y = c_a | x_i)}{\sum_{l=1}^{k} \frac{P_{D^{te}}(y=c_l)}{P_{D^{tr}}(y=c_l)} P_{tr}(y = c_l | x_i)}$$

El problema del HDy

HDy calcula histogramas independientes para cada clase



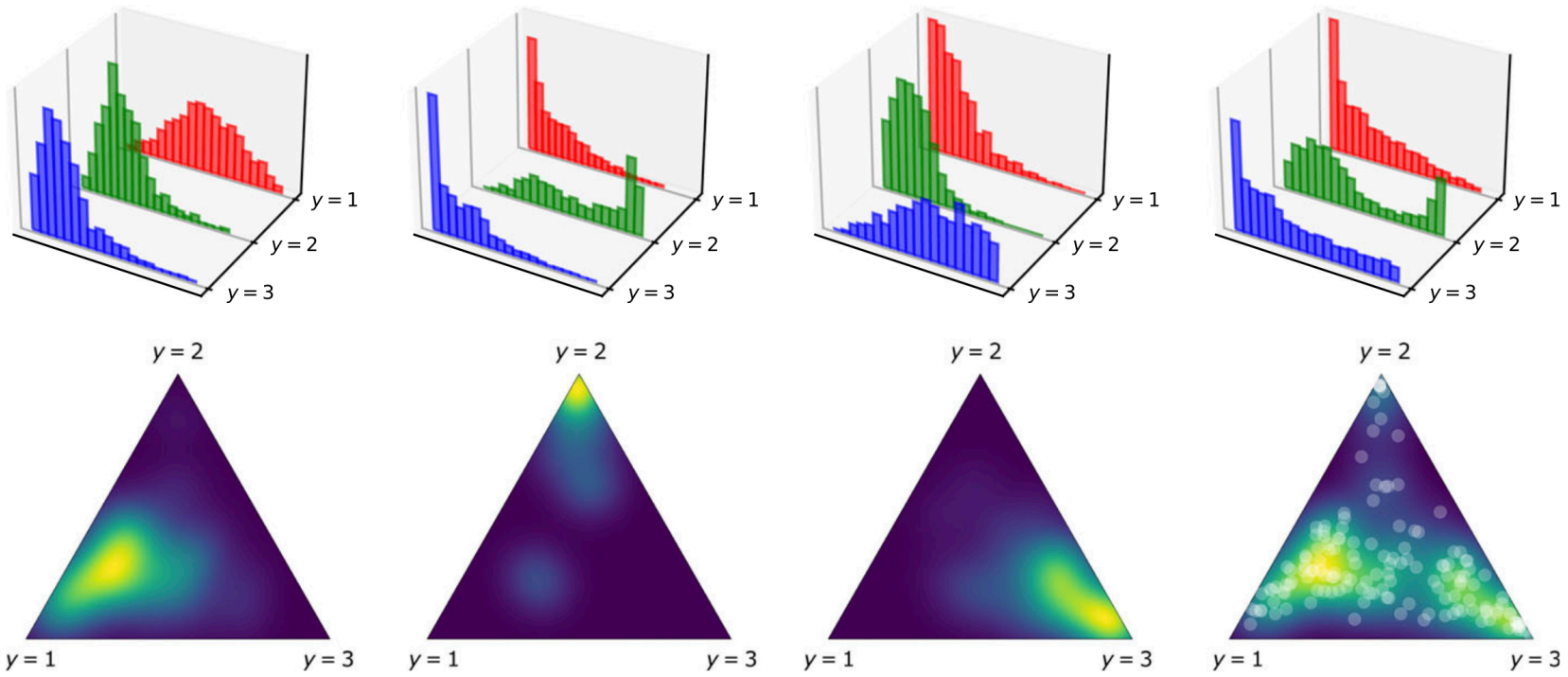
Conjuntos distintos pueden generar los mismos histogramas

$$A = \begin{Bmatrix} a_1 & = & (0{,}1, & 0{,}2, & 0{,}3, & 0{,}4) \\ a_2 & = & (0{,}2, & 0{,}3, & 0{,}4, & 0{,}1) \\ a_3 & = & (0{,}3, & 0{,}4, & 0{,}1, & 0{,}2) \end{Bmatrix} \quad A' := \begin{Bmatrix} H_1 = \text{hist}(\{0{,}1, 0{,}2, 0{,}3\}) \\ H_2 = \text{hist}(\{0{,}2, 0{,}3, 0{,}4\}) \\ H_3 = \text{hist}(\{0{,}3, 0{,}4, 0{,}1\}) \\ H_4 = \text{hist}(\{0{,}4, 0{,}1, 0{,}2\}) \end{Bmatrix}$$

$$B = \begin{Bmatrix} b_1 & = & (0{,}1, & 0{,}3, & 0{,}4, & 0{,}2) \\ b_2 & = & (0{,}3, & 0{,}2, & 0{,}1, & 0{,}4) \\ b_3 & = & (0{,}2, & 0{,}4, & 0{,}3, & 0{,}1) \end{Bmatrix} \quad B' := \begin{Bmatrix} H'_1 = \text{hist}(\{0{,}1, 0{,}3, 0{,}2\}) \\ H'_2 = \text{hist}(\{0{,}3, 0{,}2, 0{,}4\}) \\ H'_3 = \text{hist}(\{0{,}4, 0{,}1, 0{,}3\}) \\ H'_4 = \text{hist}(\{0{,}2, 0{,}4, 0{,}1\}) \end{Bmatrix}$$

KDEy [Moreo et al 2025]

- Proyectar las probabilidades sobre el simplex de dimensión nº de clases − 1
- Estimar las densidades usando Kernel Density Estimation



Alejandro Moreo, Pablo González y Juan José del Coz (2025). "Kernel density estimation for multiclass quantification". En: *Machine Learning* 114.4, pág. 92

Enfoque Simétrico

Aprendizaje Simétrico

Los métodos que hemos visto usan un **enfoque asimétrico**:

Datos, problema de clasificación: $D^{tr} = \{(x_i^{tr}, y_i^{tr})\}_{i=1}^n$

Modelo, regresión sobre conjuntos: $\bar{h} : \mathbb{N}^{\mathcal{X}} \longrightarrow [0,1]$

Sería mejor poder usar un enfoque simétrico

Datos, problema de cuantificación sobre conjuntos:

$$D^{tr} = \{(D_i^{tr}, y_i^{tr})\}_{i=1}^b, D_i^{tr} = \{x_{ij}^{tr}\}_{j=1}^n, y_i^{tr} \in [0,1]^k$$

- Ventajas
  - Se puede optimizar realmente el modelo para una función de pérdida
  - Aprende la distribución de las bags (no prior, no covariate,...)
  - Se puede usar cuando no tienes ejemplos individuales etiquetados
- Desventajas
  - Aprender sobre objetos que son conjuntos no es sencillo
  - Difícil conseguir datos así en muchos problemas reales

Deep Learning

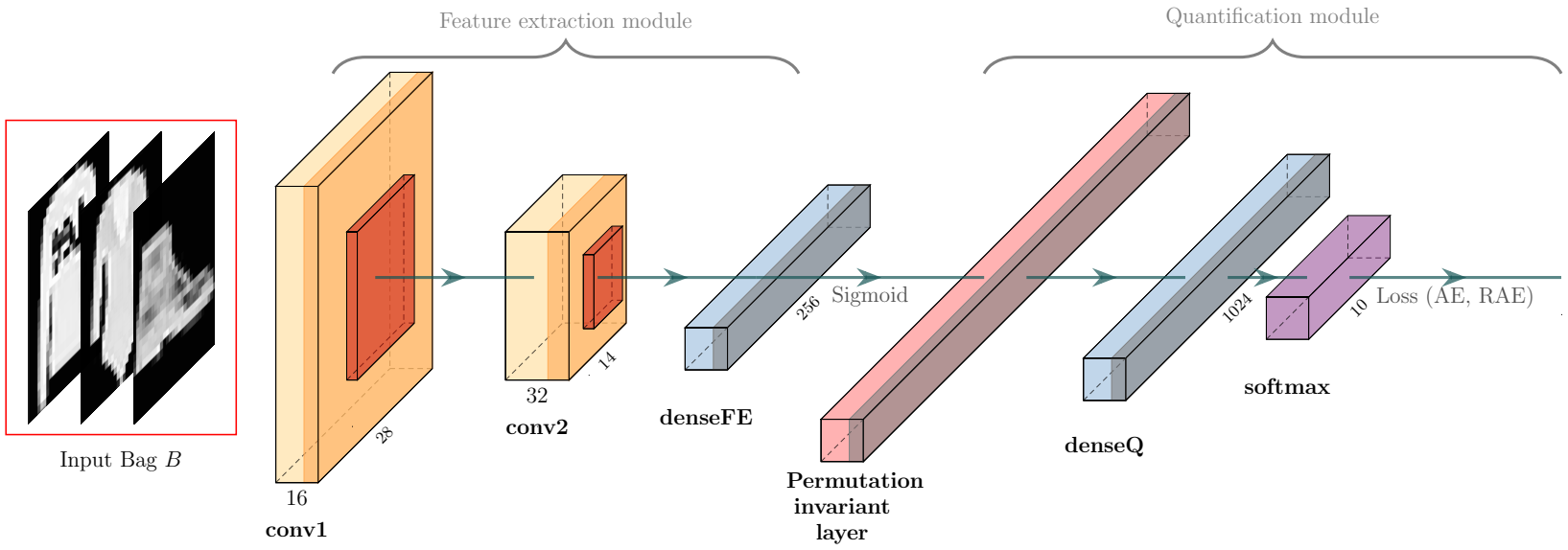
Los métodos de Deep Learning son especialmente buenos para tratar datos complejos y en cuantificación tenemos conjuntos o bags

1. Deep Sets [Zaheer et al. 2017]
   - Usan operadores de pooling que son **invariantes a las permutaciones**: máximo, media, ...
   - Proponen usar Transformers sobre conjuntos: SetTransformes
2. Deep Sets para cuantificación [Qi et al. 2020]
   - Es simplemente la aplicación de Deep Sets, con las capas de pooling como el máximo o la media

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov y Alexander J Smola (2017). "Deep sets". En: *Advances in neural information processing systems* 30

Lei Qi, Mohammed Khaleel, Wallapak Tavanapong, Adisak Sukul y David Peterson (2020). "A framework for deep quantification learning". En: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, págs. 232-248

Arquitectura



HistNetQ [Pérez-Mon et al 2025]

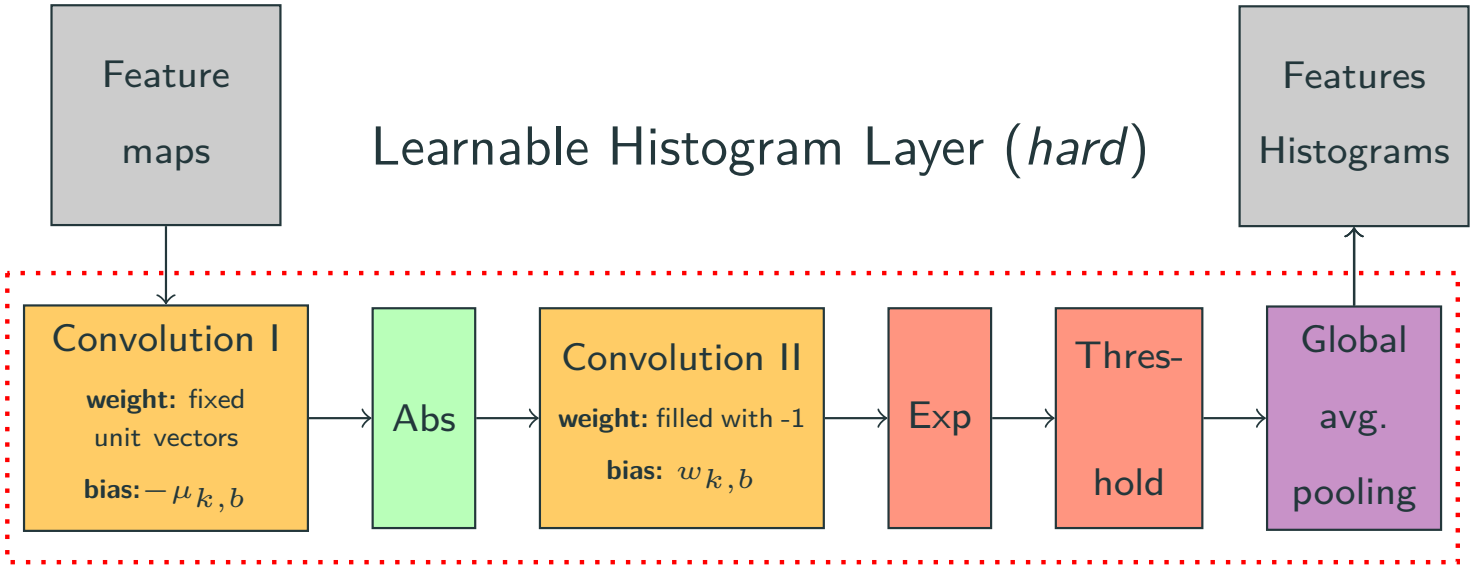
Dos elementos fundamentales:

1. Usar histogramas para representar la distribución de las bags
   - Aprende los centros y la anchura de cada bin
   - Hard binning o Soft binning
2. Generación de bags: BagMixer
   - Coge dos bags del conjunto de entrenamiento y las mezcla (mitad y mitad)
   - La prevalencia es la media. Sabemos que tiene un cierto ruido, pero es menor que si se cuantificase la muestra

Olaya Pérez-Mon, Alejandro Moreo, Juan José del Coz y Pablo González (2025). "Quantification using permutation-invariant networks based on histograms". En: *Neural Computing and Applications* 37.5, págs. 3505-3520

Hard Binning

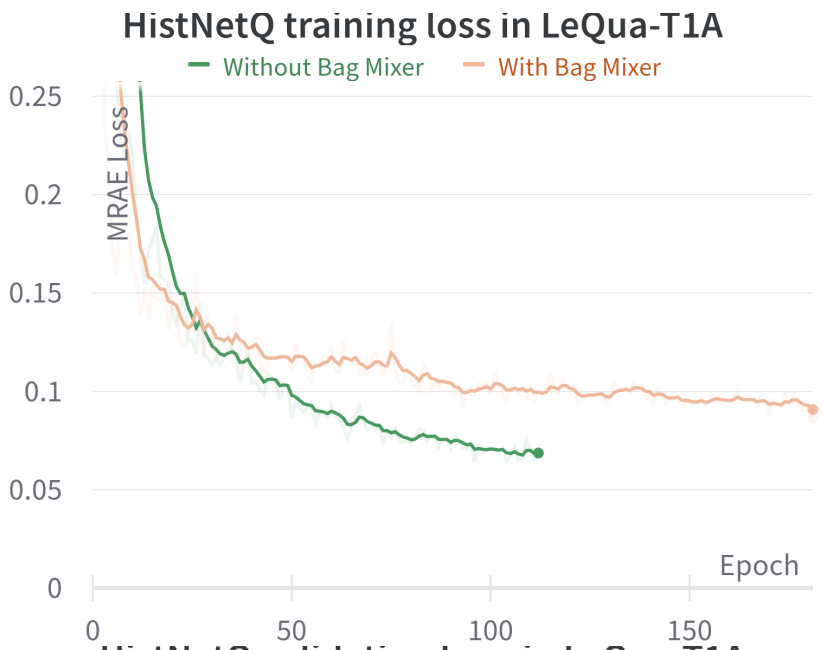
$$\phi_{k,b}(f_{k,i}) = \begin{cases} 0, & \text{si } 1{,}01^{w_{k,b} - |f_{k,i} - \mu_{k,b}|} \leq 1 \\ 1, & \text{si no} \end{cases}$$

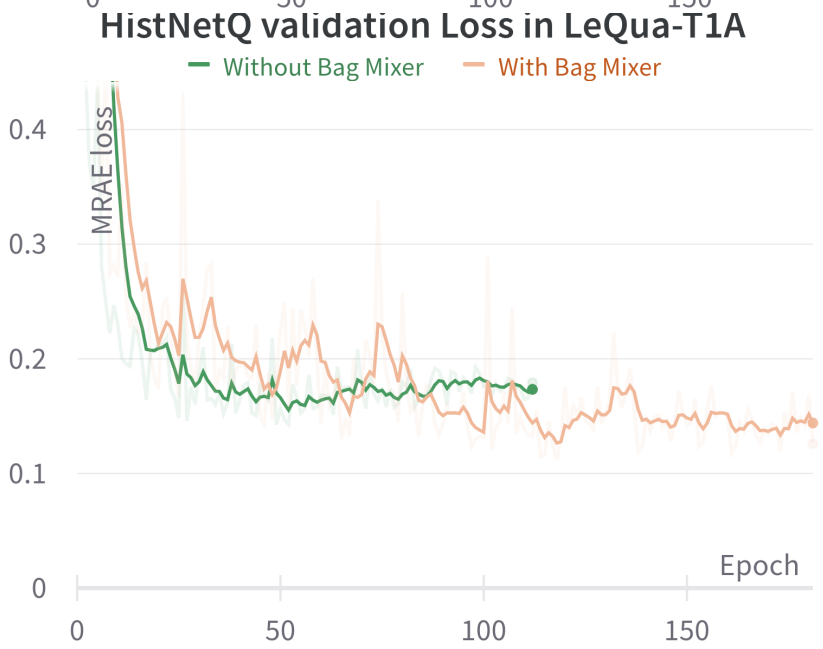


Resultados

	LeQua-T1B		LeQua-T2		Fashion-MNIST	
	AE	RAE	AE	RAE	AE	RAE
CC	0.0141 ± 0.003	1.8936 ± 1.187	0.0166 ± 0.003	2.3096 ± 1.383	0.0163 ± 0.007	0.5828 ± 0.723
PCC	0.0171 ± 0.003	2.2646 ± 1.416	0.0193 ± 0.003	2.6751 ± 1.605	0.0204 ± 0.008	0.7817 ± 0.974
ACC	0.0184 ± 0.004	1.4213 ± 1.270	0.0164 ± 0.004	1.3479 ± 1.161	0.0082 ± 0.003	0.2226 ± 0.238
PACC	0.0158 ± 0.004	1.3054 ± 0.988	0.0155 ± 0.004	1.1942 ± 1.135	0.0067 ± 0.002	0.1831 ± 0.193
EMQ-BCTS	0.0117 ± 0.003	0.9372 ± 0.817	0.0138 ± 0.004	1.1500 ± 0.978	0.0065 ± 0.002	**0.1510 ± 0.152**
EMQ-NoCalib	0.0118 ± 0.003	0.8780 ± 0.751	**0.0134 ± 0.003**	1.1616 ± 0.991	0.0132 ± 0.005	0.2549 ± 0.222
DeepSets (avg)	0.0128 ± 0.004	0.9954 ± 0.658	0.0408 ± 0.010	1.6982 ± 2.263	0.0083 ± 0.003	0.3283 ± 0.233
DeepSets (med)	0.0143 ± 0.004	0.8443 ± 0.543	0.0209 ± 0.006	1.2353 ± 0.891	0.0094 ± 0.003	0.7195 ± 0.586
DeepSets (max)	0.0277 ± 0.005	1.4646 ± 1.026	0.0219 ± 0.004	2.4217 ± 1.879	0.0219 ± 0.007	0.3520 ± 0.323
SetTransformers	0.0385 ± 0.008	1.6748 ± 1.428	0.0384 ± 0.013	3.6275 ± 4.218	0.0104 ± 0.003	2.2017 ± 1.190
HistNetQ (ours)	**0.0107 ± 0.004**	**0.7574 ± 0.489**	0.0181 ± 0.006	**0.9508 ± 0.576**	**0.0060 ± 0.002**	‡0.1592 ± 0.171

Bag Mixer


HistNetQ training loss in LeQua-T1A
Without Bag Mixer
With Bag Mixer
MRAE Loss
0.25
0.2
0.15
0.1
0.05
0
Epoch
0
50
100
150


HistNetQ validation Loss in LeQua-T1A
Without Bag Mixer
With Bag Mixer
MRAE loss
0.4
0.3
0.2
0.1
0
Epoch
0
50
100
150

Referencias

Pablo González, Alberto Castaño, Chawla Nitesh y Juan José del Coz (2017). "A Review on Quantification Learning". En: *ACM Computing Surveys* 50.2

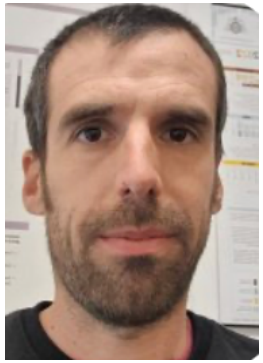
Alberto Castaño, Jaime Alonso, Pablo González, Pablo Pérez y Juan José del Coz (2024). "QuantificationLib: A Python library for quantification and prevalence estimation". En: *SoftwareX* 26, pág. 101728

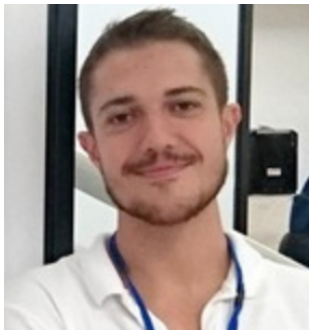
Alberto Castaño, Jaime Alonso, Pablo González y Juan José del Coz (2023). "An equivalence analysis of binary quantification methods". En: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 6, págs. 6944-6952

Alberto Castaño, Pablo González, Jaime Alonso González y Juan Jose Del Coz (2022). "Matching distributions algorithms based on the Earth mover's distance for ordinal quantification". En: *IEEE Transactions on Neural Networks and Learning Systems* 35.1, págs. 1050-1061

Olaya Pérez-Mon, Juan José del Coz y Pablo González (2025). *Quantification via Gaussian Latent Space Representations*. arXiv: 2501.13638 [cs.LG]. URL: https://arxiv.org/abs/2501.13638

Gracias!









Introducción a la Cuantificación

Estimando prevalencias por clase mediante aprendizaje supervisado

Juan José del Coz Velasco
juanjo@uniovi.es
Centro de Inteligencia Artificial - Universidad de Oviedo