UNIVERSIDAD DE BUENOS AIRES
Facultad de Ciencias Exactas y Naturales

Estimación de proporción de clases en muestras no etiquetadas mediante modelos de cuantificación

Tesis presentada para optar al título de Magister en Estadística Matemática de la Universidad de Buenos Aires

Ing. Maximiliano Marufo da Silva

Director de tesis: Dr. Andrés Farall

Buenos Aires, 2025.

Resumen

La cuantificación consiste en proporcionar predicciones agregadas para conjuntos de datos, en lugar de predicciones individuales para cada dato. En el contexto de la clasificación, esto se traduce en predecir la proporción de cada clase dentro de un conjunto de instancias, en lugar de la clase particular de cada instancia individualmente.

Un ejemplo práctico es la predicción de la proporción de comentarios positivos y negativos sobre un producto, servicio o candidato en redes sociales. Si bien se podría utilizar un clasificador para predecir el sentimiento de cada comentario y, posteriormente, derivar las proporciones de clase, esta estrategia es subóptima y a menudo produce estimaciones sesgadas de la prevalencia, lo que resulta en una baja precisión en la cuantificación. Por consiguiente, se han desarrollado métodos específicos para abordar la cuantificación como una tarea independiente.

Los modelos de cuantificación se entrenan con datos cuya distribución puede diferir de la de los datos de prueba. En el contexto de la cuantificación binaria, para cada instancia $i \in \{1, \ldots, n\}$, consideramos un vector de variables aleatorias $(\boldsymbol{X}_i, Y_i, S_i)$, donde $\boldsymbol{X}_i \in \mathbb{R}^d$ representa las características de la instancia, $Y_i \in \{0, 1\}$ denota su etiqueta de clase, y $S_i \in \{0, 1\}$ indica si la instancia está etiquetada (y, por lo tanto, pertenece al conjunto de entrenamiento). Cuando $S_i = 0$, la etiqueta Y_i no es observable. El objetivo es estimar $\theta := \mathbb{P}(Y = 1|S = 0)$, es decir, la prevalencia de etiquetas positivas entre las instancias no etiquetadas. No se asume que esta prevalencia sea igual a la de las instancias etiquetadas, $\mathbb{P}(Y = 1|S = 1)$. Además, el estimador de θ debe depender únicamente de los datos disponibles: las características de todas las instancias y las etiquetas observadas.

El objetivo de este trabajo es describir el problema de la cuantificación, justificando la necesidad de utilizar modelos optimizados para estos casos, y presentar una revisión del estado del arte en este campo, evaluando mediante simulaciones los principales modelos propuestos.

Palabras Clave: Cuantificación, estimación de proporción de clases, cambio de distribución

Abstract

Quantification aims to provide aggregate predictions for datasets, rather than individual predictions for each data point. In the context of classification, this translates to predicting the proportion of each class within a set of instances, rather than the specific class of each instance individually.

A practical example is predicting the proportion of positive and negative comments regarding a product, service, or candidate on social media. While a classifier could be used to predict the sentiment of each comment and subsequently derive class proportions, this strategy is suboptimal and often yields biased prevalence estimates, resulting in poor quantification accuracy. Consequently, dedicated methods have been developed to address quantification as an independent task.

Quantification models are trained on data whose distribution may differ from that of the test data. In the context of binary quantification, for each instance $i \in \{1, \dots, n\}$, we consider a vector of random variables $(\boldsymbol{X}_i, Y_i, S_i)$, where $\boldsymbol{X}_i \in \mathbb{R}^d$ represents the instance's features, $Y_i \in \{0, 1\}$ denotes its class label, and $S_i \in \{0, 1\}$ indicates whether the instance is labeled (and therefore belongs to the training set). When $S_i = 0$, the label Y_i is unobserved. The objective is to estimate $\theta := \mathbb{P}(Y = 1|S = 0)$, i.e., the prevalence of positive labels among unlabeled instances. This prevalence is not assumed to be equal to that of the labeled instances, $\mathbb{P}(Y = 1|S = 1)$. Furthermore, the estimator of θ must depend solely on the available data: the features of all instances and the observed labels.

This work aims to describe the quantification problem, justifying the need for optimized models in these cases, and to present a review of the state of the art in this field, evaluating the main proposed models through simulations.

Keywords: Quantification, class proportion estimation, distribution shift

Índice general

Capítulo 1

Problema

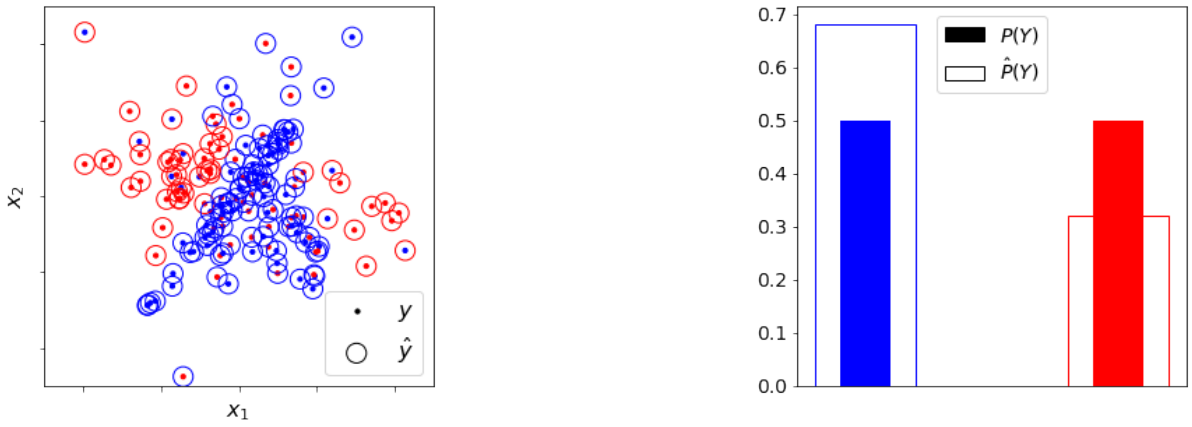
1.1. Introducción

En algunas aplicaciones vinculadas a la clasificación, el objetivo final no es determinar a qué clase (o clases) pertenece cada una de las instancias individuales de un conjunto de datos no etiquetado, sino estimar la proporción (también llamada 'prevalencia', 'frecuencia relativa' o 'probabilidad prior') de cada clase en los datos no etiquetados. En los últimos años se ha señalado que, en estos casos, tiene sentido optimizar directamente algoritmos de aprendizaje automático para este objetivo, en lugar de simplemente optimizar clasificadores para etiquetar instancias individuales.

La tarea de ajustar estimadores de prevalencia de clases a través del aprendizaje supervisado se conoce como 'aprender a cuantificar' o, más simplemente, cuantificar o *quantification* (término acuñado por Forman [1], quien planteó el problema por primera vez). Se sabe que cuantificar mediante la clasificación de cada instancia no etiquetada a través de un clasificador estándar y luego contando las instancias que han sido asignadas a cada clase (el método *Classify & Count -CC-*) generalmente conduce a estimadores de prevalencia de clases sesgados, es decir, obtienen poca exactitud en la cuantificación. Como resultado, se han desarrollado métodos que abordan la cuantificación como una tarea en sí.

Para ver la importancia de diferenciar el problema de cuantificación del de clasificación, veamos dos ejemplos. En el primero, una empresa que ofrece un servicio a sus clientes realiza una encuesta con varias preguntas para determinar el grado de satisfacción de cada persona. El objetivo de le empresa es determinar aquellos clientes que podrían no estar conformes con el servicio y ofrecerles una mejora en las condiciones para retenerlos. En el segundo ejemplo, una consultora analiza *tweets* para estimar el grado de aprobación de candidatos políticos. Aquí, la consultora no está interesado en predecir si un individuo específico está a favor o en contra, sino en cuántos encuestados, del número total de encuestados, aprueban al candidato, es decir, en conocer la prevalencia de la clase positiva.

Mientras en el primer escenario el interés es a nivel individual, en el último, el nivel agregado es lo que importa; en otras palabras, en el primer escenario la clasificación es el objetivo, mientras que en el segundo el verdadero objetivo es la cuantificación. De hecho, en la mayoría de las aplicaciones las predicciones que interesan no son a nivel individual sino a nivel colectivo; ejemplos de tales campos son la investigación de mercado, la ciencia política, las ciencias sociales, modelado ecológico y epidemiología.



(a) Clasificación

(b) Cuantificación

Figura 1.1: En la clasificación, la predicción es a nivel individual, mientras que en la cuantificación es a nivel agregado (para mayor información sobre los datos del gráfico consultar el Capítulo 2)

En resumen, y generalizando no solo para clasificación sino también a otros problemas (regresión, ordinalidad, etc.), la tarea de cuantificación consiste en proporcionar predicciones agregadas para conjuntos de datos, en vez de predicciones particulares sobre los datos individuales. Si bien en principio no es necesario realizar predicciones por cada individuo, muchos de los métodos se basan en obtener la cuantificación de esa manera, ya que hacer predicciones individuales suele ser un requisito de por sí de las aplicaciones prácticas, o porque ya existen en ellas modelos que las generen.

La literatura sobre métodos relacionados con cuantificación está un tanto desconectada. Algunos de los métodos que pueden usarse como cuantificadores han sido ideados para otros fines, principalmente para mejorar la precisión en clasificación cuando cambia el dominio. El desempeño de este último grupo ha sido normalmente estudiado solo en términos de mejora en las tareas de clasificación pero no como cuantificadores. Dado este escenario, y debido a la variedad de campos en los que ha surgido como una necesidad de aplicación, los algoritmos que se pueden aplicar para tareas de cuantificación aparecen en artículos que usan diferentes palabras clave y nombres, como *counting* [2], *prior probability shift* [3, 4], *posterior probability estimation* [5], *class prior estimation* [6–8], *class prior change* [9], *prevalence estimation* [10], *class ratio estimation* [11] o *class distribution estimation* [12–14], por citar solo algunos de ellos.

1.2. Tipos de Cuantificación

Aunque el estudio de la cuantificación se ha centrado principalmente en el dominio de clasificación, la cuantificación también aparece en otros tipos de problemas de aprendizaje automático, como la regresión, la clasificación ordinal, el aprendizaje sensible al costo y la cuantificación en redes.

De manera similar a la regresión, aprender a cuantificar admite diferentes problemas de interés aplicativo, basados en cuántas clases distintas existen en el problema, y en cuántas de las clases se pueden atribuir al mismo tiempo al mismo individuo. Así, los problemas de cuantificación se dividen de esta manera:

1. Etiquetado simple *(Single-Label Quantification -SLQ-)*: cuando cada individuo pertenece exactamente a una de las clases en $C = \{c_1, \ldots, c_{\#C}\}$.

2. Etiquetado múltiple *(Multi-Label Quantification -MLQ-)*: cuando cada individuo puede pertenecer a cualquier número de clases (cero, una o varias) en $C = \{c_1, \ldots, c_{\#C}\}$.

3. Cuantificación Binaria *(Binary Quantification -BQ-)*: se puede definir de dos formas (equivalentes) según el tipo de etiquetado definido:

   a) en *SLQ* con $\#C = 2$, (en este caso $C = \{c_1, c_2\}$, y cada individuo pertenece a c_1 o c_2)

   b) en *MLQ* con $\#C = 1$, (en este caso $C = \{c\}$, y cada individuo pertenece o no a c)

4. Cuantificación Ordinal *(Ordinal Quantification -OQ-)*: cuando existe un orden $c_1 \prec \cdots \prec c_{\#C}$ en $C = \{c_1, \ldots, c_{\#C}\}$.

5. Cuantificación de Regresión *(Regression Quantification -RQ-)*: cuando no hay un conjunto de clases involucradas, sino que cada individuo está etiquetado con una puntuación de valor real y la cuantificación equivale a estimar la fracción de ítems cuya puntuación está en un intervalo dado $[a, b]$ con $a, b \in \mathbb{R}^d$.

1.3. Marco teórico

Si hablamos entonces de cuantificación binaria, se tiene que por cada muestra $i \in \{1, \ldots, n\}$, $(\boldsymbol{X}_i, Y_i, S_i)$ es un vector de variables aleatorias tal que $\boldsymbol{X}_i \in \mathbb{R}^d$ son las características de la muestra, $Y_i \in C$ con $C = \{1, 0\}$ indica la clase a la que pertenece y $S_i \in \{1, 0\}$ indica si pertenece al conjunto de entrenamiento o al de prueba. Es decir, cuando $S_i = 0$, entonces Y_i no es observable. El objetivo es estimar $\theta := \mathbb{P}(Y = 1|S = 0)$[1], es decir, la prevalencia de etiquetas positivas en conjuntos de prueba. Esta prevalencia no se asume de ser la misma que en el conjunto de entrenamiento, $\mathbb{P}(Y = 1|S = 1)$. Además, el estimador de θ debe depender solo de los datos disponibles, es decir, de las características de la muestra (tanto del conjunto de entrenamiento como de prueba) y de las etiquetas del conjunto de entrenamiento. Los supuestos que se asumen [15] son:

- $(\boldsymbol{X}_1, Y_1, S_1) \ldots (\boldsymbol{X}_n, Y_n, S_n)$ son independientes. Es decir, el conocimiento del valor de una muestra no proporciona ninguna información sobre el valor de la otra, y viceversa.

- Por cada $s \in \{0, 1\}$, $(\boldsymbol{X}_1, Y_1)|S_1 = s, \ldots, (\boldsymbol{X}_n, Y_n)|S_n = s$ son idénticamente distribuidas. En otras palabras, que si separamos los conjuntos de entrenamiento y de prueba, en cada uno de ellos los individuos son tomados de la misma distribución de probabilidad.

- Por cada $(y_1, \ldots, y_n) \in \{0, 1\}^n$, $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n)$ es independiente de $(S_1, \ldots, S_n)$ condicionado a $(Y_1, \ldots, Y_n) = (y_1, \ldots, y_n)$. Este es el supuesto más fuerte, ya que implica que la relación entre las características de un individuo y su etiqueta no está intermediada por el conjunto (entrenamiento o prueba) al que pertenece. Es decir, que una vez que conocemos la etiqueta del individuo, sabemos la distribución de sus características, independientemente del conjunto al que pertenece el individuo.

[1] En cuantificación, se lo nombra generalmente como p (o $p_1, \ldots, p_{\#C}$ o $p(c)$ para el caso multiclase) en vez de θ, por lo que en este trabajo también se usará esta nomenclatura.

Usando la distribución de probabilidad conjunta, podemos factorizar usando las distribuciones condicionales:

$$\mathbb{P}(\boldsymbol{X}, Y, S) = \mathbb{P}(\boldsymbol{X}|Y, S)\mathbb{P}(Y|S)\mathbb{P}(S) \tag{1.1}$$

Luego, usando el tercer supuesto mencionado, podemos hacer [3]:

$$\mathbb{P}(\boldsymbol{X}, Y, S) = \mathbb{P}(\boldsymbol{X}|Y)\mathbb{P}(Y|S)\mathbb{P}(S) \tag{1.2}$$

Si bien existen varios métodos propuestos para el aprendizaje de cuantificación [16, 17], el mismo es todavía relativamente desconocido incluso para expertos en aprendizaje automático. La razón principal es la creencia errónea de que es una tarea trivial que se puede resolver usando un método directo, como *CC*. La cuantificación requiere métodos más sofisticados si el objetivo es obtener modelos óptimos, y su principal dificultad radica en la definición del problema, ya que las distribuciones de los datos de entrenamiento y de prueba pueden ser distintas. Por ejemplo, si la diferencia entre $\mathbb{P}(Y = 1|S = 0)$ y $\mathbb{P}(Y = 1|S = 1)$ es grande, los métodos simples como *CC* suelen tener bajo rendimiento.

1.4. Cambios en las distribuciones de los datos

En los últimos años ha habido un interés creciente en las aplicaciones que presentan cambios en las distribuciones de datos (conocido en la bibliografía por su término en inglés *dataset shift*). Estos problemas comparten el hecho de que la distribución de los datos utilizados para entrenar es diferente a la de los datos que se usan para predecir. Al igual que para el área de la cuantificación, aquí también la literatura sobre el tema está dispersa y diferentes autores usan diferentes nombres para referirse a los mismos conceptos, o usan el mismo nombre para diferentes conceptos.

Teniendo en cuenta que en los problemas de clasificación tenemos:

- Un conjunto de características o covariables $\boldsymbol{X}$.
- Una variable de respuesta Y.
- Una distribución de probabilidad conjunta $\mathbb{P}(Y = y, \boldsymbol{X} = \boldsymbol{x})$.

La probabilidad conjunta $\mathbb{P}(Y, \boldsymbol{X})$ luego se puede escribir como $\mathbb{P}(Y|\boldsymbol{X})\mathbb{P}(\boldsymbol{X})$ o como $\mathbb{P}(\boldsymbol{X}|Y)\mathbb{P}(Y)$. Por otro lado, cuando usamos los términos de entrenamiento (*train*) y prueba (*test*), nos referimos a las datos disponibles para entrenar al clasificador y los datos presentes en el entorno en el que se implementará el clasificador, respectivamente. Podemos entonces también separar los datos en dos distribuciones distintas, condicionando a la variable S definida en 1.3, siendo $\mathbb{P}_{tr}(Y, \boldsymbol{X}) = \mathbb{P}(Y, \boldsymbol{X}|S = 1)$ y $\mathbb{P}_{tst}(Y, \boldsymbol{X}) = \mathbb{P}(Y, \boldsymbol{X}|S = 0)$.

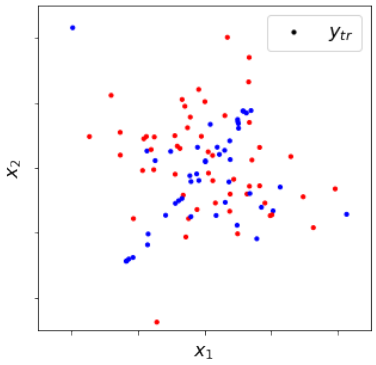
El *dataset shift* aparece cuando las distribuciones conjuntas de entrenamiento y de prueba son diferentes, es decir, cuando $\mathbb{P}_{tr}(Y, \boldsymbol{X}) \neq \mathbb{P}_{tst}(Y, \boldsymbol{X})$. Moreno-Torres et al. [3] distingue las distintas variantes del *dataset shift* según qué elementos mencionados anteriormente cambian:

- *Covariate shift*, cuando $\mathbb{P}_{tr}(Y|\boldsymbol{X}) = \mathbb{P}_{tst}(Y|\boldsymbol{X})$ y $\mathbb{P}_{tr}(\boldsymbol{X}) \neq \mathbb{P}_{tst}(\boldsymbol{X})$
- *Prior probability shift*, cuando $\mathbb{P}_{tr}(\boldsymbol{X}|Y) = \mathbb{P}_{tst}(\boldsymbol{X}|Y)$ y $\mathbb{P}_{tr}(Y) \neq \mathbb{P}_{tst}(Y)$
- *Concept shift*, cuando $\mathbb{P}_{tr}(Y|\boldsymbol{X}) \neq \mathbb{P}_{tst}(Y|\boldsymbol{X})$ y $\mathbb{P}_{tr}(\boldsymbol{X}) = \mathbb{P}_{tst}(\boldsymbol{X})$ o $\mathbb{P}_{tr}(\boldsymbol{X}|Y) \neq \mathbb{P}_{tst}(\boldsymbol{X}|Y)$ y $\mathbb{P}_{tr}(Y) = \mathbb{P}_{tst}(Y)$

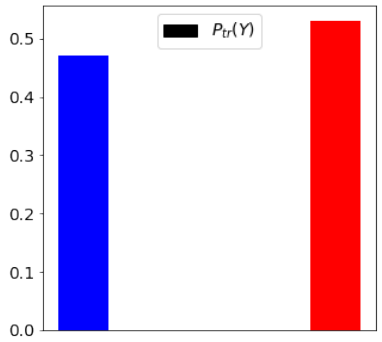
Otros tipos de *dataset shift* surgen cuando $\mathbb{P}_{tr}(Y|\boldsymbol{X}) \neq \mathbb{P}_{tst}(Y|\boldsymbol{X})$ y $\mathbb{P}_{tr}(\boldsymbol{X}) \neq \mathbb{P}_{tst}(\boldsymbol{X})$ y cuando $\mathbb{P}_{tr}(\boldsymbol{X}|Y) \neq \mathbb{P}_{tst}(\boldsymbol{X}|Y)$ y $\mathbb{P}_{tr}(Y) \neq \mathbb{P}_{tst}(Y)$. Sin embargo, estos tipos de cambios no se consideran generalmente en la literatura ya que aparecen mucho más raramente, o incluso porque son difíciles o imposibles de resolver.

El problema de cuantificación se trata de un caso donde $\mathbb{P}_{tst}(Y)$ es desconocido. Además, la mayoría de los métodos de cuantificación propuestos asumen que $\mathbb{P}_{tr}(\boldsymbol{X}|Y) = \mathbb{P}_{tst}(\boldsymbol{X}|Y)$, por lo que están dentro de los casos de *prior probability shift*. Esto implica que se asumen los tres supuestos mencionados en 1.3.

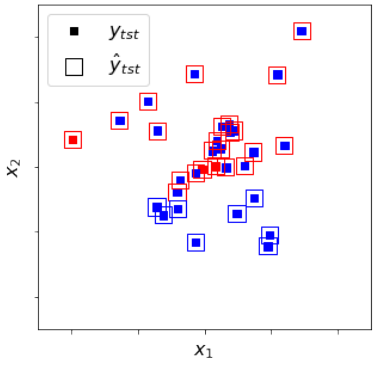
Por otro lado, en la mayoría de los casos el objetivo final de la implementación es estimar algún parámetro de $\mathbb{P}_{tst}(Y)$. Por ejemplo, como ya mencionamos anteriormente, en la cuantificación binaria, se desea estimar $\theta := \mathbb{P}(Y = 1|S = 0)$, o lo que es lo mismo, $p_{tst} := \mathbb{P}_{tst}(Y = 1)$. Es decir, en la cuantificación la tarea indirectamente suele ser aprender a aproximar una distribución desconocida (observando solo características de una muestra) mediante una distribución conocida. En consecuencia, prácticamente todas las medidas de evaluación para la cuantificación son divergencias, es decir, medidas de cómo una distribución pronosticada difiere de la distribución real.



(a) Muestra de entrenamiento

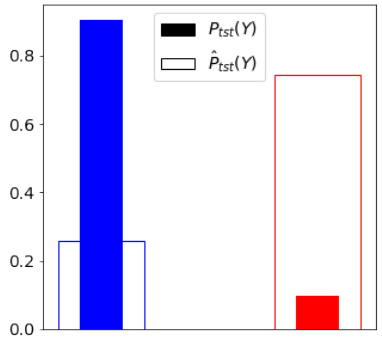


(b) Prevalencia de clases en muestra de entrenamiento



(c) Muestra de prueba[a] y su clasificación

[a] con $\mathbb{P}_{tst}(\boldsymbol{X}|Y) = \mathbb{P}_{tr}(\boldsymbol{X}|Y)$



(d) Prevalencia de clases verdadera y cuantificación en muestra de prueba

Figura 1.2: El *prior probability shift* propio de los problemas de cuantificación puede hacer que los métodos simples de cuantificación, como *CC*, tengan grandes errores.

1.5. El problema de clasificar y contar

En ausencia de métodos para estimar los valores de prevalencia de clase de forma directa, el primer método que suele pensarse para hacerlo es *Classify & Count* o *CC*, es decir, clasificar cada individuo no etiquetado y estimar los valores de prevalencia de clase contando los individuos que fueron asignados a cada clase. Sin embargo, esta estrategia es subóptima: si bien un clasificador perfecto produce también un cuantificador perfecto, un buen clasificador puede producir un peor cuantificador que el que produce un mal clasificador. Para ver esto, se puede ver la definición de F_1, una función de evaluación estándar para la clasificación binaria, que se define como:

$$F_1 = \frac{2 \cdot tp}{2 \cdot tp + fp + fn} \tag{1.3}$$

donde tp, fp, fn indican el número de verdaderos positivos, falsos positivos y falsos negativos, respectivamente. Un buen clasificador, según la métrica F_1, puede producir un mal cuantificador ya que F_1 considera buenos aquellos clasificadores que mantienen la suma $fp + fn$ al mínimo; sin embargo, el objetivo de un algoritmo de cuantificación debe ser mantener al mínimo $|fp - fn|$. Es decir, que un mal clasificador con fp y fn altos pero similares podría llegar a producir un mejor cuantificador que un buen clasificador.

El análisis teórico de esta cuestión se basa en el supuesto de *prior probability shift* 1.4. Bajo tal supuesto, la estimación $\hat{p}$ obtenida por el enfoque *CC* depende solo de las características del clasificador, definido (para el caso binario) por su tasa de verdaderos positivos (tpr), su tasa de falsos positivos (fpr) y de la prevalencia real (p):

$$\hat{p}(p) = p \cdot tpr + (1 - p) \cdot fpr \tag{1.4}$$

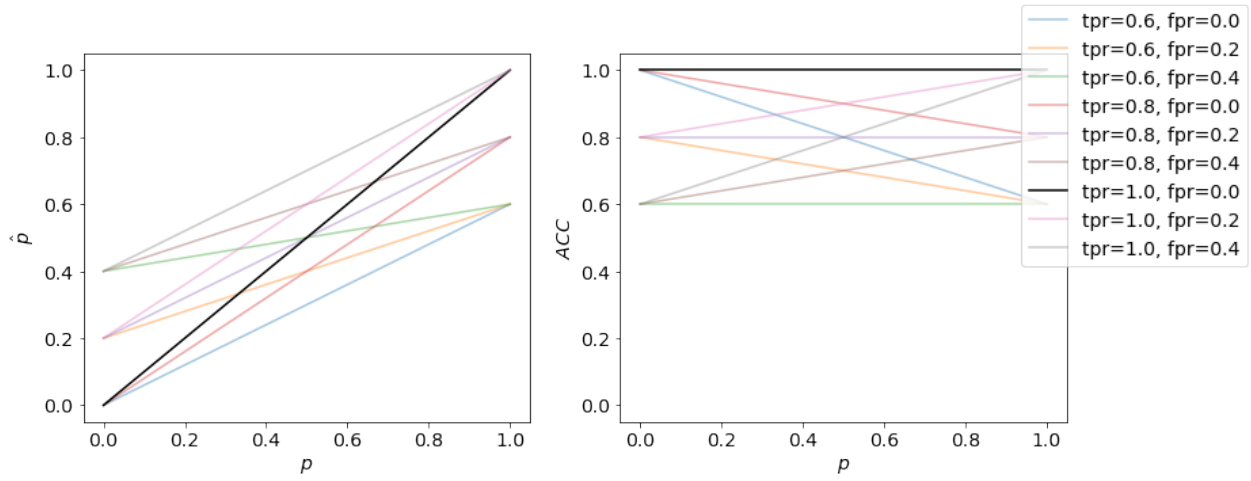


Figura 1.3: La línea negra representa el cuantificador y clasificador perfecto, respectivamente. Las otras líneas muestran las estimaciones teóricas de $\hat{p}$ resultantes de aplicar la ecuación 1.4, y el *accuracy* correspondiente al clasificador, según se varían los valores de tpr y fpr.

El mal desempeño de *CC* fue demostrado mediante el siguiente teorema por Forman [18]:

Teorema 1.1 (Toerema de Forman)**.** [18, p.169] Para un clasificador imperfecto, el método *CC* subestimará la proporción verdadera de ejemplos positivos p en un conjunto de prueba para $p > p^*$,

y sobreestimará para $p < p^*$, donde p^* es la proporción particular en la cual el método CC estima de forma correcta. Es decir, el método CC estima exactamente p^* para un conjunto de prueba con p^* muestras positivas.

La demostración (ver todos los detalles en [18, p.170]) supone que el cuantificador CC produce una predicción perfecta para una prevalencia concreta, llamada p^*, y estudia su comportamiento cuando la prevalencia cambia ligeramente. Suponiendo que $\hat{p}(p^*) = p^*$, cuando la prevalencia cambia en una cantidad $\Delta \neq 0, p^* + \Delta$, la estimación del método CC en tal caso será:

$$\begin{aligned} \hat{p}(p^* + \Delta) &= (p^* + \Delta) \cdot tpr + (1 - (p^* + \Delta)) \cdot fpr \\ &= \hat{p}(p^*) + (tpr - fpr) \cdot \Delta \\ &= p^* + (tpr - fpr) \cdot \Delta \end{aligned} \tag{1.5}$$

La predicción del método CC será perfecta, $\hat{p}(p^* + \Delta) = (p^* + \Delta)$, solo cuando el clasificador también es perfecto ($tpr = 1$, $fpr = 0$ y por tanto $tpr - fpr = 1$). Pero en el caso habitual, en el cual el clasificador es imperfecto ($0 \leq tpr - \text{fpr} < 1$), cuando la prevalencia aumenta ($\Delta > 0$), CC la subestima ($\hat{p} < p^* + \Delta$), y cuando la prevalencia disminuye ($\Delta < 0$), CC la sobreestima ($\hat{p} > p^* + \Delta$).

Un buen clasificador puede estar sesgado, es decir, puede mantener sus falsos positivos al mínimo solo a expensas de una cantidad sustancialmente mayor de falsos negativos (o viceversa); si este es el caso, el clasificador es un mal cuantificador. Este fenómeno no es infrecuente, especialmente en presencia de datos desbalanceados. En tales casos, los algoritmos que minimizan las funciones de pérdida de clasificación (*Hamming*, *hinge*, etc.) suelen generar clasificadores con tendencia a elegir la clase mayoritaria, lo que implica un número mucho mayor de falsos positivos que de falsos negativos para la clase mayoritaria, lo que significa a su vez que tal algoritmo tenderá a subestimar las clases minoritarias.

Los argumentos anteriores indican que no se debe considerar la cuantificación como un mero subproducto de la clasificación, y debe estudiarse y resolverse como una tarea en sí misma. Hay al menos otros dos argumentos que apoyan esta idea. Uno es que las funciones que se utilizan para evaluar la clasificación no se pueden utilizar para evaluar la cuantificación, ya que estas funciones miden, en general, cuántos individuos han sido mal clasificados, y no cuánto difiere la prevalencia de clase estimada del valor real. Esto significa que los algoritmos que minimizan estas funciones están optimizados para la clasificación, y no para la cuantificación. Un segundo argumento presentado por Forman [18] es que los métodos diseñados específicamente para cuantificar requieren menos datos de entrenamiento para alcanzar la misma precisión de cuantificación que los métodos estándar basados en CC. Si bien esta observación es de naturaleza empírica, también existen argumentos teóricos que sustentan este hecho [19].

1.6. Cuantificadores para la mejora de la clasificación

Debido a los problemas mencionados anteriormente de los clasificadores frente a cambios en las distribuciones de los datos y frente a datos desbalanceados, los algoritmos de cuantificación están cada vez más frecuentemente siendo usados en tareas que requieren predicciones individuales. Los mismos se emplean como suplemento de clasificadores para suplir sus defectos frente a estos

problemas, ya que en algunos casos no solo predicen los valores agregados, sino que también mejoran las predicciones a nivel individual.

Por ejemplo, el *prior probability shift* puede hacer que los clasificadores performen de manera subóptima. En el caso del clasificador óptimo de Bayes, dado por:

$$h(\boldsymbol{x}) = \arg\max_y p_{Y|\boldsymbol{X}=\boldsymbol{x}}(y) = \arg\max_y \frac{p_{\boldsymbol{X}|Y=y}(\boldsymbol{x})p_Y(y)}{p_{\boldsymbol{X}}(\boldsymbol{x})} \tag{1.6}$$

la decisión del clasificador depende de $p_Y(y)$, que es estimado con el dataset de entrenamiento, siendo $\hat{p}_Y(y=1) = p_{tr}$. Es decir, que en caso de $\mathbb{P}_{tr}(Y) \neq \mathbb{P}_{tst}(Y)$, la decisión final del clasificador puede verse afectada negativamente. Para mejorar el rendimiento del clasificador frente a estos casos, se debería usar $\hat{p}_Y(y=1) = p_{tst}$, pero como p_{tst} es generalmente desconocido, se puede usar un método de cuantificación para estimarlo [5, 8, 14, 20].

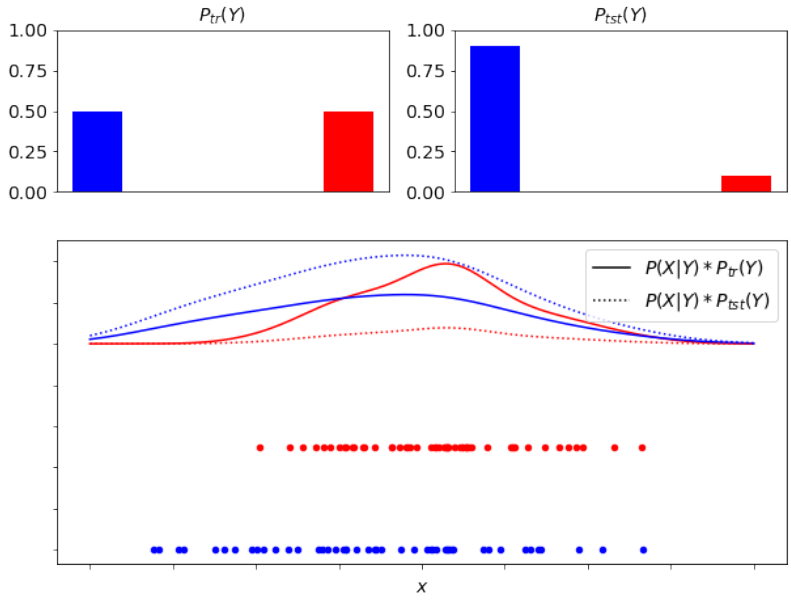


Figura 1.4: Ejemplo de como el *prior probability shift* puede alterar las predicciones del clasificador óptimo de Bayes. Vemos que si usáramos $\mathbb{P}_{tst}(Y)$ en vez de $\mathbb{P}_{tr}(Y)$, en este caso pasaría a predecir la clase azul en vez de roja para toda $\boldsymbol{x}$.

Los métodos de cuantificación pueden usarse no solo para mejorar el rendimiento general de un clasificador, sino también para mejorar su equidad o *fairness* [21, 22], es decir, su posibilidad de predecir resultados independientes de un cierto conjunto de variables que consideramos sensibles y no relacionadas con él (e.j.: género, etnia, orientación sexual, etc.). Por ejemplo, suponiendo que una variable Z debe considerarse sensible, se puede estimar $\mathbb{P}_{tr}(Y|Z)$. Luego, si los datos de entrenamiento están sesgados, por ejemplo, con $\mathbb{P}_{tr}(Y=1|Z=1) \gg \mathbb{P}_{tr}(Y=1|Z=0)$, pero se sabe que en las muestras a inferir esto no es así, se puede optimizar el modelo de clasificación imponiendo alguna penalidad basada en la estimación de $\mathbb{P}_{tst}(Y|Z)$, siendo esta última obtenido por un cuantificador.

Capítulo 2

Métodos de Estimación

Durante los últimos años, se han propuesto varios métodos de cuantificación desde diferentes perspectivas y con diferentes objetivos. En términos generales, se pueden distinguir dos grandes clases de métodos en la literatura. La primera clase es la de métodos agregativos, es decir, métodos que requieren la clasificación de todos los individuos como un paso intermedio. Dentro de los métodos agregativos, se pueden identificar dos subclases. La primera subclase incluye métodos basados en clasificadores de propósito general; en estos métodos la clasificación de los elementos individuales realizados como un paso intermedio puede lograrse mediante cualquier clasificador. La segunda subclase se compone, en cambio, de métodos que para clasificar los individuos, se basan en métodos de aprendizaje diseñados con la cuantificación en mente. La segunda clase es la de métodos no agregativos, es decir, métodos que resuelven la tarea de cuantificación "holísticamente", es decir, sin clasificar a los individuos. La idea de esta tesis no es la de mostrar todos los métodos propuestos hasta la actualidad, sino la de mencionar los métodos más populares.

Caso de Ejemplo

Como ejemplo de muestra para comparar los distintos métodos de estimación, se usará el mismo set de datos artificiales que en las figuras 1.1 y 1.2. Para crear el set de datos se utilizó el algoritmo prepuesto por Guyon [23] mediante la función *make_classification* de *scikit-learn*, usando los siguientes parámetros de entrada:

```
population_size = 150 X, y = make_classification( n_samples=
   population_size,
n_features=2, n_informative=2, n_redundant=0, n_repeated=0,
n_clusters_per_class=2, n_classes=2, weights=None, class_sep=0.1,
random_state=42, )
```

Esta función, al usar $weights = None$, crea un set de datos balanceados en cuanto a las clases de los individuos (figura 1.1). Sin embargo, para poder evaluar los distintos métodos, lo que haremos ahora es seleccionar 100 individuos y usarlos como datos de entrenamiento, y hacer un sub-muestreo de los 50 restantes de forma tal de aproximarnos a un $p_{tst} = 0.1$:

```
train_size = 100 prev_test = 0.1 X_train, y_train = X[:train_size],
y[:train_size] X_test, y_test = X[train_size:], y[train_size:]
```

```
idx_negatives_test = np.argwhere(y_test==0).flatten() idx_positives_test =
np.random.choice( np.argwhere(y_test==1).flatten(),
size=round((prev_test)*len(idx_negatives_test)/(1-prev_test)), replace=
    False )
idx_test = np.concatenate([idx_positives_test, idx_negatives_test]) X_test
    =
X_test[idx_test] y_test = y_test[idx_test]
```

El código de arriba termina generando un set de entrenamiento de tamaño $n_{tr} = 100$ con $p_{tr} = 0.53$ (figuras 1.2a y 1.2b) y uno de prueba de tamaño $n_{tst} = 31$ con $p_{tst} \approx 0.097$ (figuras 1.2c y 1.2d).

El modelo de clasificación utilizado para generar las predicciones mostradas en ambas figuras 1.1 (entrenando y prediciendo con todos los datos) y 1.2 (entrenando con los datos de entrenamiento y prediciendo los de prueba) es el clasificador *Naive Bayes* generado mediante la clase *GaussianNB* de *scikit-learn*, usando sus parámetros por default.

2.1. Métodos Agregativos

2.1.1. Con clasificadores generales

Dentro de los métodos agregativos, algunos de ellos requieren como entrada las etiquetas de clases predichas (es decir, clasificadores duros), otros requieren un *score* de decisión (como podría ser la distancia al hiperplano de separación en el clasificador SVM), y otros que requieren como entrada las probabilidades *a posteriori* de pertenencia a cada clase (es decir, clasificadores blandos)[1]. En estos últimos, además, las probabilidades *a posteriori* deben estar calibradas (para mayor información sobre calibración consultar el Apéndice A). Para estos casos, en los ejemplos a continuación que requieren clasificadores blandos se separó de las muestras de entrenamiento un 15 % de datos para el proceso de calibración (y estratificando para que la proporción de etiquetas se mantenga igual).

Clasificar y Contar (CC)

El método más sencillo y directo para construir un cuantificador para clasificación (tanto binaria como multiclase) es aplicar el enfoque *Classify & Count* [1]. *CC* juega un papel importante en la investigación de cuantificación ya que siempre se utiliza como el *baseline* que cualquier método de cuantificación razonable debe mejorar. Este método consiste simplemente en: (i) ajustar un clasificador duro, y luego (ii), utilizando dicho clasificador, clasificar las instancias de la muestra de prueba, contando la proporción de cada clase. Generalizando el estimador de *CC* para el caso multiclase, el mismo queda entonces definido por:

$$\hat{p}^{CC}_{tst}(c) = \frac{\#\{\boldsymbol{x} \in \boldsymbol{X}_{tst} | h_{tr}(\boldsymbol{x}) = c\}}{\#\boldsymbol{X}_{tst}} \tag{2.1}$$

donde se usó h_{tr} para la función de decisión del clasificador duro ajustado con la muestra de entrenamiento.

[1] Los clasificadores blandos y de *score* se pueden convertir en duros usando umbrales de clasificación

Es evidente que podemos obtener un cuantificador perfecto si el clasificador es también perfecto. El problema es que obtener un clasificador perfecto es casi imposible en aplicaciones reales, y luego el cuantificador hereda el sesgo del clasificador. Este aspecto se analiza en varios artículos tanto desde una perspectiva teórica como práctica, como lo hizo Forman [18], y como también ya lo hemos mencionado en 1.5.

Ejemplo: Para el caso de ejemplo, y manteniendo el clasificador allí usado, debemos contar la cantidad de predicciones positivas (rojas) en la figura 1.2, y dividirlas por el tamaño de la muestra de prueba. Es decir, $\hat{p}_{tst}^{CC}(c=1) = \frac{23}{31} \approx 0.742$.

Clasificar, Contar y Ajustar (ACC)

Conocido en inglés como *Adjusted Classify & Count*, *Adjusted Count* [18] o también como *Confusion Matrix Method* [20], este método se basa en corregir las estimaciones de CC teniendo en cuenta la tendencia del clasificador a cometer errores de cierto tipo. Un modelo ACC está compuesto por dos elementos: un clasificador duro (como en CC) y de las estimaciones de tpr y fpr. Dichas estimaciones pueden obtenerse usando validación cruzada o *cross-validation*, ya sea mediante la técnica de *k-folds* o un *held-out*. Luego, en la fase de predicción, el modelo obtiene una primera estimación $\hat{p}$ de la misma forma que en CC que luego, para el caso binario, es ajustado aplicando la siguiente fórmula[2]:

$$\hat{p}_{tst}^{ACC}(c=1) = \frac{\hat{p}_{tst}^{CC}(c=1) - \hat{fpr}}{\hat{tpr} - \hat{fpr}} \tag{2.2}$$

Esta expresión se obtiene despejando la verdadera prevalencia p de la ecuación 1.4 y reemplazando fpr y tpr por sus estimadores.

El método ACC es teóricamente perfecto, independientemente de la métrica de *accuracy* obtenida con el clasificador, cuando se cumple el supuesto de *prior probability shift* 1.4 y cuando las estimaciones de tpr y fpr son perfectas. Desafortunadamente, es raro que se cumplan ambas condiciones en aplicaciones del mundo real: $\mathbb{P}(\boldsymbol{X}|Y)$ puede tener variaciones entre los datos de entrenamiento y los de predicción, y es difícil obtener estimaciones perfectas para tpr y fpr en algunos dominios ya que suelen haber pequeñas muestras disponibles y/o están muy desequilibradas. Pero incluso en estos casos, el rendimiento del método ACC suele ser mejor que el de CC.

Partiendo de la ecuación 2.1 y utilizando el teorema de probabilidad total, podemos extender la ecuación 2.2 para el caso multiclase:

$$\begin{aligned} \hat{p}_{tst}^{CC}(c=c_k) &= \hat{\mathbb{P}}_{tst}(h_{tr}(\boldsymbol{x}) = c_k) \\ &= \sum_{j=1}^{\#C} \hat{\mathbb{P}}(h_{tr}(\boldsymbol{x}) = c_k | y = c_j) \hat{p}_{tst}^{ACC}(c=c_j) \end{aligned} \tag{2.3}$$

donde $\hat{p}_{tst}^{CC}(c=c_k)$ es la fracción de datos de tst que el clasificador h asigna a c_k (y por ende, es conocido), y $\hat{\mathbb{P}}(h_{tr}(\boldsymbol{x}) = c_k | y = c_j)$ es la estimación de probabilidad de que el clasificador h asigne la clase c_k a $\boldsymbol{x}$ cuando este pertenece a la clase c_j. Estas probabilidades, al igual que tpr y fpr en el caso binario, deben estimarse mediante validación cruzada [1, 10, 18]. Luego, $\hat{p}_{tst}^{ACC}(c=c_j)$, nuestras incógnitas (una por cada c_j), pueden calcularse mediante un sistema de ecuaciones lineales con $\#C$ ecuaciones y $\#C$ incógnitas.

[2] A veces, esta expresión conduce a un valor inválido de $\hat{p}_{tst}^{ACC}$ que debe recortarse en el rango $[0, 1]$ en un último paso.

Ejemplo: Aquí debemos estimar el tpr y fpr. Para ello, se separó de la muestra de entrenamiento un 15 % de datos. Con el 85 % de la muestra se entrenó el clasificador y se obtuvo un $\hat{p}_{tst}^{CC}(c=1) \approx 0.71$, y con el 15 % separado se obtuvo $\hat{tpr} \approx 0.625$ y $\hat{fpr} \approx 0.714$, y por lo tanto, $\hat{p}_{tst}^{ACC}(c=1) \approx 0.0516$.

Clasificar y Contar Probabilístico (PCC)

Este método, conocido en inglés como *Probabilistic Classify and Count* [24, 25], es una variante de *CC* que utiliza un clasificador blando en vez de uno duro. Es decir, que la salida del clasificador blando ajustado con la muestra de entrenamiento, $s(\boldsymbol{x}, y)$, será una estimación de la probabilidad *a posteriori* $p_{Y|\boldsymbol{X}=\boldsymbol{x}}(y)$ por cada individuo $\boldsymbol{x} \in \boldsymbol{X}_{tst}$ y cada $y \in C$. El método consiste en estimar las $p_{tst}(c = c_j)$ mediante el valor esperado de la proporción de items que se predijeron como pertenecientes a cada clase c_j:

$$\begin{aligned} \hat{p}_{tst}^{PCC}(c = c_j) &= \hat{\mathbb{E}}[p_{Y|\boldsymbol{X}=\boldsymbol{x}}(y = c_j)] \\ &= \frac{1}{m} \sum_{i=1}^{m} \hat{p}_{Y|\boldsymbol{X}=\boldsymbol{x}_i}(y = c_j) \\ &= \frac{1}{m} \sum_{i=1}^{m} s(\boldsymbol{x}_i, y = c_j) \end{aligned} \tag{2.4}$$

con $m = \#\boldsymbol{X}_{tst}$. La intuición detrás de *PCC* es que las probabilidades *a posteriori* contienen mayor información que las decisiones de un clasificador duro y, por lo tanto, deberían ser usadas en su lugar. Sin embargo, Tasche [26, Corolario 6, p.157 y p.163] demuestra que el comportamiento de *PCC* será similar al de *CC*, en cuanto a que ambos subestiman o sobreestiman la prevalencia verdadera cuando la distribución de clases cambia entre los datos de entrenamiento y de prueba.

Ejemplo: Como este método utiliza un clasificador blando, se separó primero un 15 % de los datos de entrenamiento. Con el 85 % se entrenó el clasificador, y luego con el 15 % se realizó la calibración. Luego, debemos sumar las salidas del clasificador calibrado para la clase positiva. Para el ejemplo, se obtuvieron las siguientes salidas:

$s(\boldsymbol{x}_i, y = 1)$	0.54	0.54	0.49	0.50	0.45	0.56	0.57	0.60	0.52	0.50	0.58	0.54 ...	0.49

siendo $\hat{p}_{tst}^{PCC}(c = 1) \approx 0.527$.

Clasificar, Contar y Ajustar Probabilístico (PACC)

Presentado como *Probabilistic Adjusted Classify and Count* o también como *Probabilistic Adjusted Count*, este método combina las ideas de ACC y de PCC [24, 25].

$$\begin{aligned}\hat{p}_{tst}^{PCC}(c=c_k) &= \hat{\mathbb{E}}[\mathbb{P}_{tst}(h_{tr}(\boldsymbol{x})=c_k)] \\ &= \hat{\mathbb{E}}[\sum_{j=1}^{\#C}\mathbb{P}(h_{tr}(\boldsymbol{x})=c_k|y=c_j)p_{tst}^{PACC}(c=c_j)] \\ &= \sum_{j=1}^{\#C}\hat{\mathbb{E}}[\mathbb{P}(h_{tr}(\boldsymbol{x})=c_k|y=c_j)p_{tst}^{PACC}(c=c_j)] \\ &= \sum_{j=1}^{\#C}\hat{\mathbb{E}}[\mathbb{P}(h_{tr}(\boldsymbol{x})=c_k|y=c_j)]\hat{p}_{tst}^{PACC}(c=c_j) \\ &= \sum_{j=1}^{\#C}[\frac{1}{\#U_j}\sum_{\boldsymbol{x}\in U_j}\hat{\mathbb{P}}(h_{tr}(\boldsymbol{x})=c_k)]\hat{p}_{tst}^{PACC}(c=c_j)\end{aligned} \tag{2.5}$$

donde $U_j = \{(\boldsymbol{x}, y) \in (\boldsymbol{X}_{tst}, Y_{tst})|y = c_j\}$. Luego, $\hat{p}_{tst}^{PCC}(c = c_k)$ se calcula mediante PCC y, como en ACC, las $[\frac{1}{\#U_j}\sum_{\boldsymbol{x}\in U_j}\hat{\mathbb{P}}(h_{tr}(\boldsymbol{x}) = c_k)]$ deben estimarse mediante validación cruzada, quedando nuevamente un sistema de ecuaciones lineales de $\#C$ ecuaciones y $\#C$ incógnitas.

Para el caso particular binario, y relacionando con la ecuación 2.2, tenemos:

$$\hat{p}_{tst}^{PACC}(c=1) = \frac{\hat{p}_{tst}^{PCC}(c=1) - \hat{fp}_{pa}}{\hat{tp}_{pa} - \hat{fp}_{pa}} \tag{2.6}$$

donde tp_{pa} y fp_{pa} (*pa*: *probability average*) son los dos parámetros propios del cuantificador a estimar mediante validación cruzada, siendo tp_{pa} el promedio de las probabilidades *a posteriori* para la clase positiva estimadas por el clasificador correspondientes a los individuos cuya etiqueta es positiva, y del mismo modo fp_{pa} pero para individuos con etiqueta negativa. En este método hay que tener en cuenta ambas consideraciones sobre las estimaciones de $\hat{p}$ dentro del rango $[0, 1]$ y sobre la calibración -ver A-.

Ejemplo: Del mismo modo que para el ejemplo de PCC, se separó de la muestra de entrenamiento un 15 % de datos para realizar la calibración del clasificador blando. Pero también se separó otro 15 % para realizar el ajuste del propio método de cuantificación. Con el 70 % de datos se entrenó el clasificador que luego fue calibrado usando el primer 15 % separado, obteniendo un $\hat{p}_{tst}^{PCC}(c = 1) \approx 0.51$. Luego, con el segundo 15 % de datos, se procedió a estimar tp_{pa} y fp_{pa}. Teniendo en cuenta entonces ahora tanto las salidas del clasificador calibrado como las etiquetas de la muestra, tenemos:

$s(\boldsymbol{x}_i, y = 0)$	$s(\boldsymbol{x}_i, y = 1)$	c
0.23	0.77	1
0.78	0.22	0
0.40	0.60	1
0.43	0.57	1
0.32	0.68	1
...		
0.58	0.42	0

siendo entonces $\hat{tp}_{pa} \approx 0.551$ y $\hat{fp}_{pa} \approx 0.548$, por lo que $\hat{p}_{tst}^{PACC}(c = 1) \approx 1.00$ (teniendo que haber truncado).

Selección de Umbrales (TH)

Cuando los datos de entrenamiento presentan un desbalance significativo (generalmente los casos positivos son los escasos), la precisión de ACC se ve considerablemente afectada [27]. En estas situaciones, el clasificador tiende a favorecer la predicción de la clase mayoritaria (negativa), lo que disminuye la cantidad de fp pero a expensas de un bajo tpr. Esto se traduce en un denominador reducido en la ecuación 2.2, lo que hace que el método sea más sensible a las estimaciones de tpr y fpr.

Esta serie de métodos se fundamenta en la elección de un umbral que reduzca la varianza en las estimaciones de tpr y fpr. La premisa es identificar un umbral que aumente el número de tp, aunque generalmente esto conlleve un incremento fpr. Siempre que $tpr \gg fpr$, el denominador en 2.2 aumenta, lo que resulta en métodos más robustos ante pequeños errores en las estimaciones de tpr y fpr. Siguiendo esta lógica, Forman [18, 27] propone una serie de métodos basados en clasificadores que entreguen *scores* (no necesariamente probabilísticos ni calibrados) con distintas estrategias de selección de umbrales[3]:

- MAX: selecciona el umbral que maximiza $tpr - fpr$. Esto resulta en el mayor denominador posible en la ecuación 2.2 para el clasificador entrenado, lo que suaviza las correcciones.
- X: busca obtener $fpr = 1 - tpr$ para evitar los extremos de ambas curvas.
- T50: elige el umbral con $tpr = 0.5$, asumiendo que los positivos conforman la clase minoritaria. El objetivo es nuevamente evitar los extremos de la curva tpr.
- Median Sweep (MS): adopta un enfoque conjunto, calculando la prevalencia para todos los umbrales que modifiquen los posibles valores de fpr y tpr, y devolviendo la mediana de estas prevalencias como la predicción final.

Ejemplo: En la siguiente figura se visualiza la selección de umbral según los criterios MAX, X y T50. Con estos umbrales, se computa luego la etapa de clasificación y, utilizando los correspondientes fpr y tpr, se utiliza la ecuación 2.2:

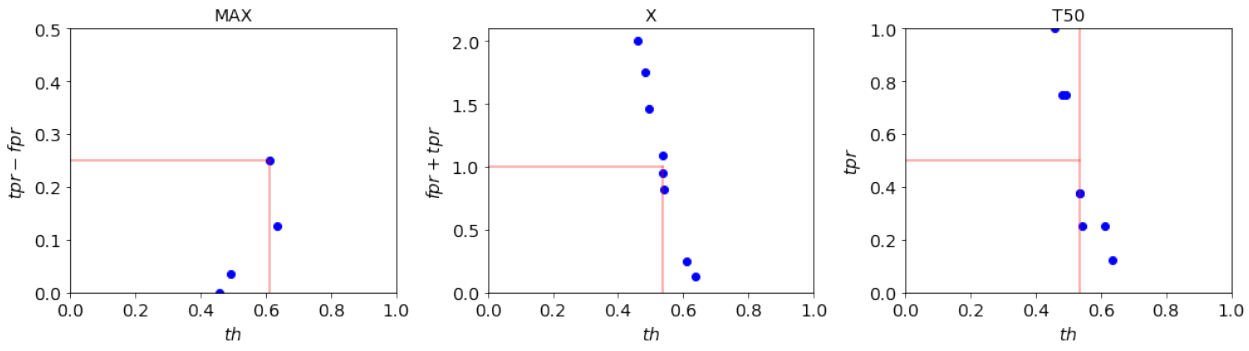


Figura 2.1

[3]Los métodos aquí se describen son exclusivamente de cuantificación binaria (las versiones multiclase no han sido abordadas en la literatura y no son sencillas de implementar)

Para el criterio MS, en cambio, por cada umbral que cambie fpr o tpr se calcula una prevalencia (se descartan los casos indeterminados por 2.2), y luego la mediana de todas ellas será la predicción final del método.

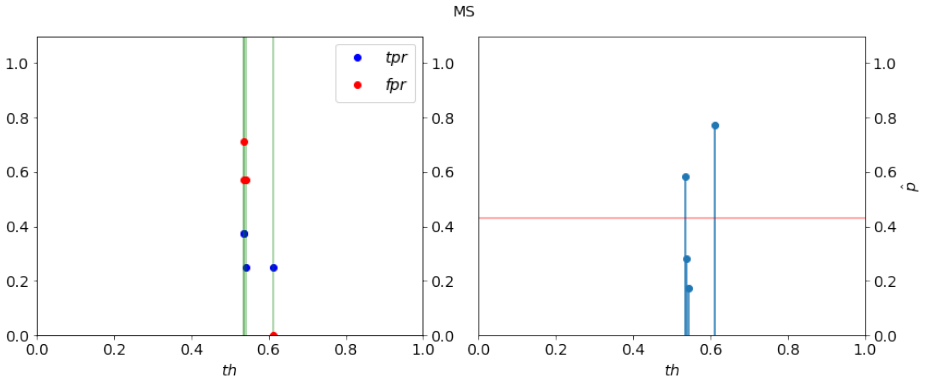


Figura 2.2

Los resultados obtenidos fueron:

- $\hat{p}_{tst}^{MAX}(c=1) \approx 0.774$
- $\hat{p}_{tst}^{X}(c=1) \approx 0.282$
- $\hat{p}_{tst}^{T50}(c=1) \approx 0.282$
- $\hat{p}_{tst}^{MS}(c=1) \approx 0.433$

Esperanza-Maximización (EMQ)

Aunque este método se propuso originalmente para mejorar las probabilidades *a posteriori* de modelos de clasificación bajo *dataset shift* (ver 1.4), el mismo también sirve para mejorar la estimación de prevalencias. También conocido como *SLD* por las iniciales de sus autores, este método fue propuesto por Saerens et al. [20] y aplica el algoritmo de Esperanza-Maximización (EM) [28], un conocido algoritmo iterativo para encontrar estimaciones de máxima verosimilitud de parámetros (los valores de prevalencia de clase) para modelos que dependen de variables no observadas (las etiquetas de clase). Esencialmente, *EMQ* actualiza incrementalmente las probabilidades *a posteriori* utilizando los valores de prevalencia de clases calculados en el último paso de la iteración, y actualiza los valores de prevalencia de clases utilizando las probabilidades *a posteriori* calculadas en el último paso de la iteración, de forma mutuamente recursiva, y tomando como punto de partida un valor determinado para la prevalencia de clases (generalmente el valor correspondiente a la muestra de entrenamiento o una estimación *a priori* dada por algún conocimiento de la muestra de prueba, aunque puede ser cualquier otro valor), y repitiendo las iteraciones hasta alcanzar la convergencia.

Saerens et al. [20, Apéndice, p.23 a p.25] demuestra, mediante el Teorema de Bayes y el Teorema de probabilidad total, que el algoritmo de EM aplicado a este problema resulta en los siguientes pasos (el paso 0 se aplica una sola vez, luego se iteran el E y M):

0 - Inicialización de $\hat{p}_Y^{(0)}(y=c_k)$, generalmente haciendo $\hat{p}_Y^{(0)}(y=c_k) = \hat{p}_{tr}(c=c_k)$

E - Esperanza:
$$\hat{p}^{(s)}_{Y|\boldsymbol{X}=\boldsymbol{x}_i}(y=c_k)=\frac{\frac{\hat{p}^{(s)}_Y(y=c_k)}{\hat{p}_{tr}(c=c_k)}s(\boldsymbol{x}_i,y=c_k)}{\sum_{j=1}^{\#C}\frac{\hat{p}^{(s)}_Y(y=c_j)}{\hat{p}_{tr}(c=c_j)}s(\boldsymbol{x}_i,y=c_k)}$$

M - Maximización:
$$\hat{p}^{(s+1)}_Y(y=c_k)=\frac{1}{m}\sum_{i=1}^{m}\hat{p}^{(s)}_{Y|\boldsymbol{X}=\boldsymbol{x}_i}(y=c_k)$$

Finalmente, cuando se alcanza la convergencia, se obtiene: $\hat{p}^{EMQ}_{tst}(c=c_k)=\hat{p}_Y(y=c_k)$.

Aunque ya mencionamos que el modelo supone que las probabilidades *a posteriori* de modelos de clasificación ya están calibradas, se ha estudiado también que el método *EMQ* mejora las predicciones de cuantificación si el clasificador utilizado está calibrado [29, 30].

Ejemplo: Comenzamos con la inicialización, dando en nuestro caso como resultado $\hat{p}^{(0)}_Y(y=1)=\hat{p}_{tr}(c=1)=0.5$ y $\hat{p}^{(0)}_Y(y=0)=\hat{p}_{tr}(c=0)=0.5$. En la primera iteración, para el paso E queda $\hat{p}^{(s=0)}_{Y|\boldsymbol{X}=\boldsymbol{x}_i}(y=c_k)=s(\boldsymbol{x}_i,y=c_k)$ es decir, las mismas salidas del clasificador calibrado. Luego, para el paso M, y al igual que en *PCC*, debemos promediar cada salida individual del clasificador calibrado por cada una de las clases existentes, quedando en nuestro caso $\hat{p}^{(s=1)}_Y(y=1)=p^{PCC}_{tst}(c=1)\approx 0.527$. Ahora, en el paso E de la segunda iteración, se usará este último valor junto con los valores de prevalencia de la muestra de entrenamiento para ajustar las salidas del clasificador calibrado. Por ejemplo, para ajustar la salida del primer individuo correspondiente a la clase positiva, sería: $\hat{p}^{(s=1)}_{Y|\boldsymbol{X}=\boldsymbol{x}_i}(y=1)\approx\frac{\frac{0.527}{0.5}0.539}{\frac{0.527}{0.5}0.539+\frac{0.473}{0.5}0.461}$. Si continuamos repitiendo los pasos E y M de forma sucesiva, y definiendo un criterio de corte para la convergencia (ya sea por máxima cantidad de iteraciones o por un umbral de diferencia entre $\hat{p}^{(s)}_Y(y=c_k)$ y $\hat{p}^{(s+1)}_Y(y=c_k)$), se obtuvo $p^{EMQ}_{tst}(c=1)\approx 0.164$.

Usando la distancia de Hellinger en y (HDy)

González-Castro et al. [12] proponen dos métodos fundamentados en la comparación de distribuciones. Aunque difieren en la manera de representar estas distribuciones, ambos comparten un elemento esencial: emplean la distancia de Hellinger como medida para cuantificar la disparidad entre ellas. El primer método, conocido como *HDy*, es un método agregativo ya que emplea las salidas del clasificador para describir las distribuciones tanto de la muestra de entrenamiento como la de prueba. El método se basa en el cálculo de:

$$p^{HDy}_{tst}(c=1)=\underset{0\leq\alpha\leq 1}{\arg\min}\,\mathrm{HD}(\alpha f_{tr}(s(\boldsymbol{x},y=1|c=1))+(1-\alpha)f_{tr}(s(\boldsymbol{x},y=1|c=0)),f_{tst}(s(\boldsymbol{x},y=1))) \tag{2.7}$$

donde:

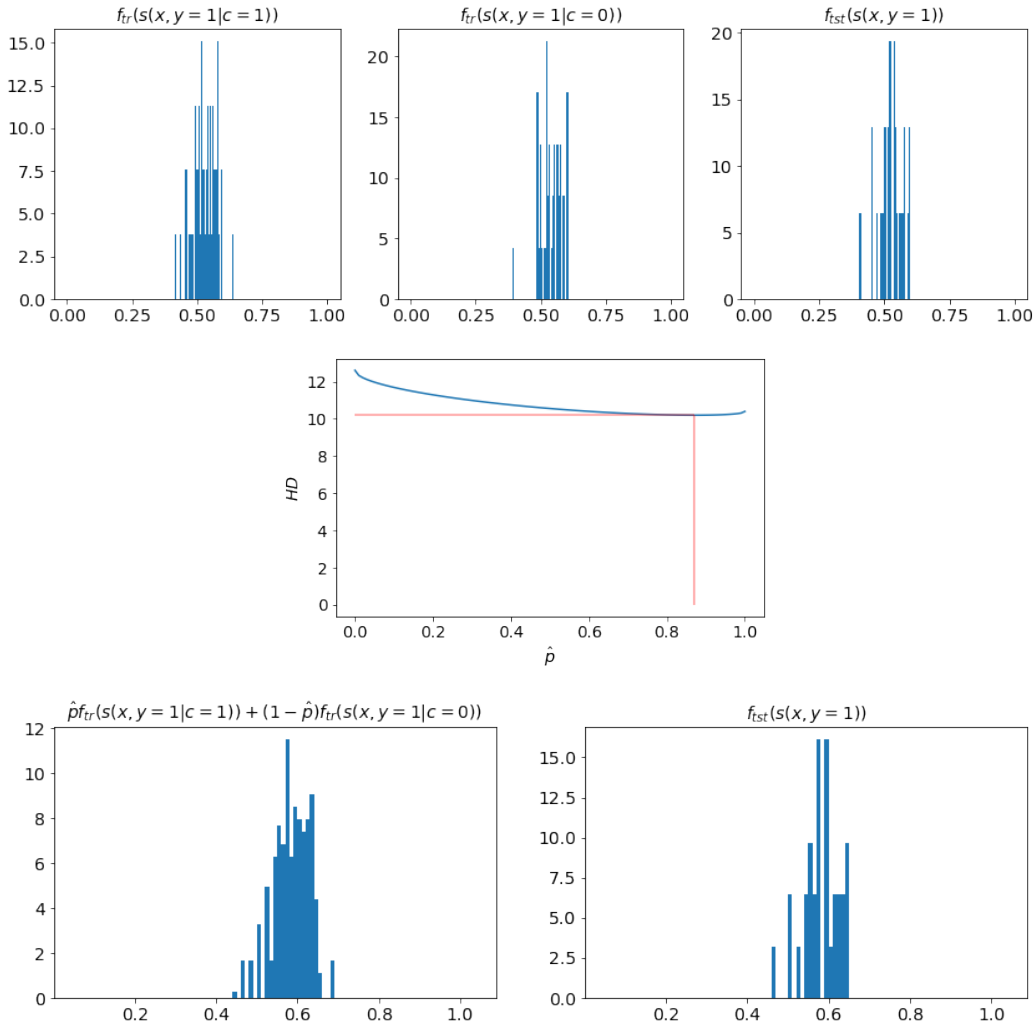
$$\mathrm{HD}(P\parallel Q)=\frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{k}\left(\sqrt{p_i}-\sqrt{q_i}\right)^2}\ \text{con}\ P=(p_1,\ldots,p_k),Q=(q_1,\ldots,q_k) \tag{2.8}$$

y $f_{tr}(s)$ y $f_{tst}(s)$ son las funciones de densidad de probabilidad de las salidas del clasificador para la muestra de entrenamiento y de evaluación, respectivamente. Estas densidades son aproximadas empíricamente mediante histogramas, siendo k el número de *bins* utilizados. Dado que el número de

bins k podría tener un impacto significativo en la estimación, normalmente se utiliza como estimador la mediana de la distribución de los α encontrados para un rango de k.

El segundo método propuesto por González-Castro et al. [12] pertenece a los métodos no agregativos y será desarrollado en la correspondiente sección 2.2.

Ejemplo: En las siguientes gráficas vemos cómo son las estimaciones de tres distribuciones estimadas para el ejemplo, la gráfica de la función de costo usada, y cómo el mínimo encontrado se usa para combinar las distribuciones de entrenamiento y compararlas con la de prueba. En este caso, se utilizó sólo $k = 200$.



Se observa que el valor mínimo de HDy se da en $p_{tst}^{HDy}(c = 1) = 0.87$.

2.1.2. Con clasificadores específicos

Los métodos presentados anteriormente implican el uso de un clasificador, a menudo seguido por una fase de ajuste para contrarrestar cualquier tendencia del clasificador a subestimar o sobreestimar las proporciones de clases. Los algoritmos discutidos en esta sección están específicamente diseñados con este propósito en mente: durante el entrenamiento, tienen en cuenta que el modelo será utilizado para cuantificar.

Minimización de pérdida explícita (ELM)

Esta familia de métodos se aplican en principio a la cuantificación binaria, pero son fácilmente extensibles a la cuantificación multiclase. La idea propuesta por Esuli y Sebastiani [31] es seleccionar una medida de rendimiento de cuantificación y entrenar un algoritmo de optimización para construir el modelo óptimo según esa medida. Las diferencias entre ellos se deben a la medida de rendimiento seleccionada y al algoritmo de optimización utilizado.

Esuli y Sebastiani [31–33] proponen utilizar SVM_{perf} [34] para optimizar la divergencia KL -ver 3.2-, mientras que Barranquero et al. [35] también emplean SVM_{perf} pero con una pérdida diferente, argumentando que la cuantificación pura no considera la precisión del clasificador subyacente (pudiendo generar un modelo que, aunque cuantifique bien, clasifique mal). Para abordar esto, introducen la medida Q, que combina una métrica de cuantificación con una métrica de clasificación, permitiendo un equilibrio entre ellas. Más recientemente Moreo y Sebastiani [36] reincorporaran la idea de utilizar SVM_{perf}, pero sugieren usar las métricas de error absoluto (AE) y error absoluto negativo (RAE) -ver 3.2-.

Existen dos inconvenientes asociados con SVM_{perf}: podría resultar en un modelo menos óptimo y no escala para grandes cantidades de datos de entrenamiento. Para abordar estas limitaciones, Kar et al. [37] proponen algoritmos de optimización estocástica. Además, plantean distintas métricas multivariadas para evaluar el rendimiento de cuantificación. Siguiendo esta línea, Sanyal et al. [38] introducen una serie de algoritmos que permiten el entrenamiento directo de redes neuronales profundas y la generación de clasificadores no lineales. Estos métodos están diseñados para optimizar funciones de pérdida de cuantificación como la divergencia KL.

Ejemplo: A diferencia de los casos anteriores, aquí no usamos el mismo clasificador con el que veníamos trabajando. En cambio, se ajusta un nuevo clasificador pero con una función de pérdida más acorde al problema de cuantificación. A modo de ejemplo, usaremos el método propuesto por Esuli et al. [31], es decir, usando la divergencia KL como pérdida y SVM_{perf} como algoritmo de optimización. De esta forma, obtuvimos un $p_{tst}^{SVM_{perf},KLD}(c=1) \approx 0.613$, lo cual efectivamente implica una mejora del KLD con respecto a usar el clasificador anterior con CC, ya que $KLD_{tst}^{CC} \approx 0.887$ y $KLD_{tst}^{SVM_{perf},KLD} \approx 0.556$.

2.2. Métodos No Agregativos

Hasta ahora, hemos utilizado métodos que agregan predicciones individuales de un clasificador para poder cuantificar. Sin embargo, también es posible estimar valores de prevalencia de clase sin generar decisiones binarias o probabilidades *a posteriori* para cada ítem. Esta alternativa se fundamenta en el principio de Vapnik, que sugiere resolver problemas directamente con la información disponible en lugar de abordar un problema más general. En cuantificación, esto significa que podemos estimar prevalencias de clase directamente sin clasificar cada individuo.

Usando la distancia de Hellinger en $\boldsymbol{x}$ (HDx)

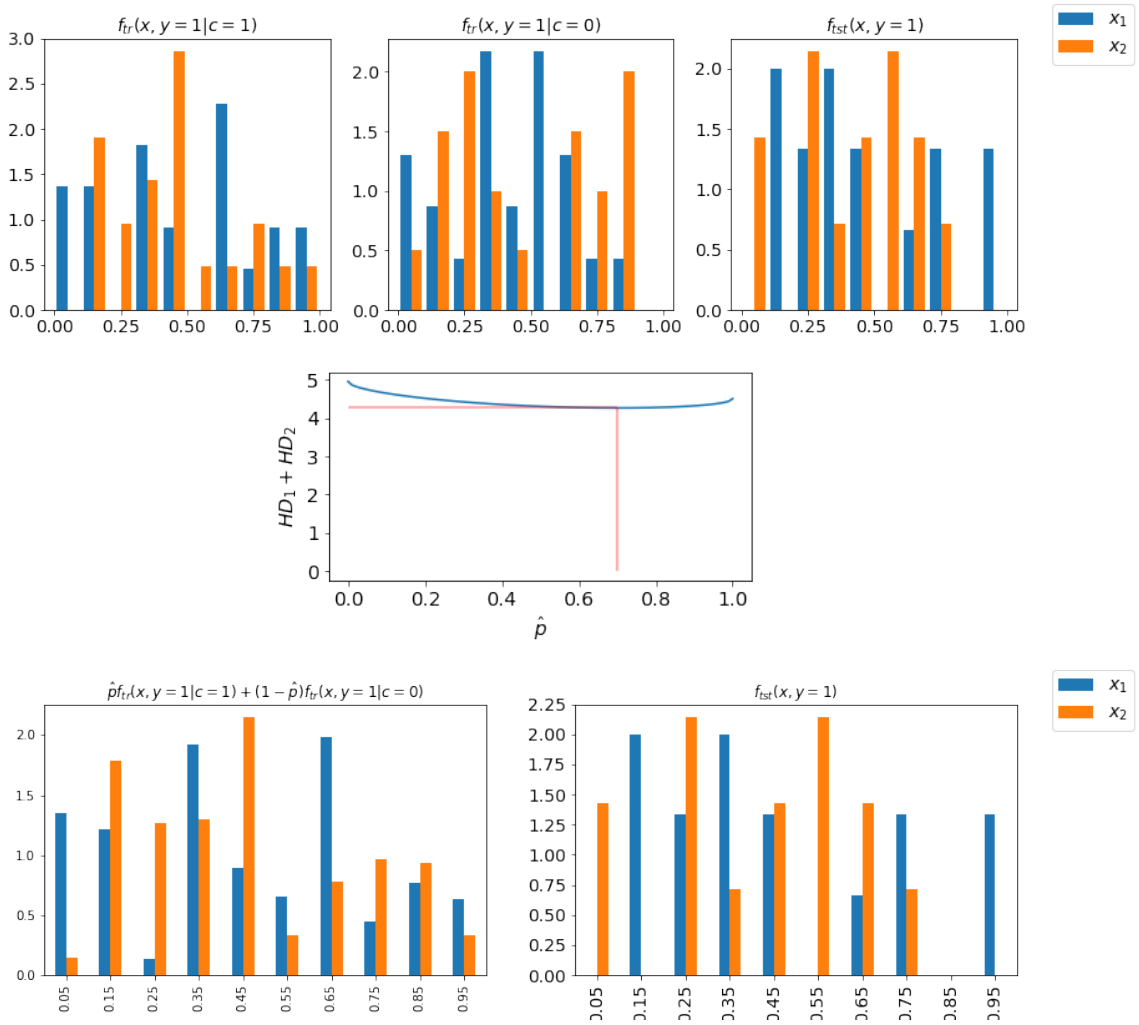
Este método está obviamente relacionado con *HDy* (2.1.1), con la diferencia de considerar distribuciones de probabilidad multidimensionales $f(\boldsymbol{x})$ en lugar de distribuciones unidimensionales

$f(s(\boldsymbol{x}))$. En vez de utilizar las salidas del clasificador, se estiman, con los datos de entrenamiento, las funciones densidad de las características de los individuos condicionadas a sus etiquetas.

Debido a la multidimensionalidad de $\boldsymbol{x} \in \mathbb{R}^d$, González-Castro et al. [12] proponen minimizar el promedio de las divergencias de Hellinger por cada $\boldsymbol{x}^j$:

$$p_{tst}^{HDx}(c=1) = \underset{0\leq\alpha\leq1}{\arg\min} \frac{1}{d}\sum_{j=1}^{d} \mathrm{HD}(\alpha f_{tr}(\boldsymbol{x^j}|c=1) + (1-\alpha) f_{tr}(\boldsymbol{x^j}|c=0), f_{tst}(\boldsymbol{x^j})) \tag{2.9}$$

Ejemplo: Repetimos el tipo de gráficos mostrados para caso de *HDy*, siendo en este caso $k = 10$.



Se observa que el valor mínimo de HDx se da en $p_{tst}^{HDx}(c=1) = 0.7$.

Usando modelos generativos

Keith and O'Connor [39] presentan un enfoque con modelos generativos para estimar la prevalencia. Este método realiza una inferencia directa de la prevalencia desconocida y aborda además el cálculo de intervalos de confianza (IC)[4].

[4]Este enfoque es pionero en la cuantificación, ya que introduce un modelo directo para el cálculo de los IC, los cuales eran estimados mediante *bootstraping* en trabajos anteriores

Su propuesta, inspirada también en Saerens et al. [20], se basa en modelar la distribución conjunta de las características de los individuos y sus etiquetas, tanto en las datos de entrenamientos como en los de evaluación. En base a esto se computa la verosimilitud marginal sobre $\theta = p_{tst}$ para obtener la distribución *a posteriori* de θ:

$$MLL_{tst}(\theta) = \sum \log \sum_{c \in C} p_{tst}(\boldsymbol{x}_i, y_i = c|\theta) \tag{2.10}$$

donde analizando para el caso binario, teniendo en cuenta que se asume $p_{tst}(\boldsymbol{x}) = p_{tr}(\boldsymbol{x})$, y utilizando las funciones de densidad de probabilidad de las características condicionadas a las etiquetas estimadas con los datos de entrenamiento ($p_{tr}(\boldsymbol{x}|y)$), podemos escribir:

$$MLL_{tst}(\theta) = \sum \log \sum_{c \in C} \theta p_{tr}(\boldsymbol{x}_i|y_i = 1) + (1 - \theta) p_{tr}(\boldsymbol{x}_i|y_i = 0) \tag{2.11}$$

Luego, para obtener la predicción se obtiene el máximo de la distribución. Es decir, que al igual que el método *EMQ*, se busca maximizar la verosimilitud, pero en este caso no necesariamente utilizando el algoritmo *EM*. Esta función es unimodal en $\theta \in [0, 1]$. Como es cóncava y hay un sólo parámetro, se pueden emplear muchas técnicas para encontrar la moda, incluyendo *EM*, Newton-Rapshon o computacionalmente mediante una grilla de valores.

Keith and O'Connor [39] proponen particularmente dos modelos generativos enfocados en problemas de procesamiento de lenguaje (*MNB y Loglin*), al que llaman explícitos. Pero aún más interesante es el tercer método que proponen, al que llaman *LR-Implicit*, el cual se basa en estimar de forma implícita las $p(\boldsymbol{x}|y)$ que obtendríamos con modelos generativos mediante las $p(y|\boldsymbol{x})$ que se obtienen con modelos discriminativos, utilizando el Teorema de Bayes:

$$p_{disc}(y|\boldsymbol{x}) = \frac{p_{imp}(\boldsymbol{x}|y) p_{tr}(y)}{p(\boldsymbol{x})} \tag{2.12}$$

Siendo $p_{disc}(y|\boldsymbol{x}) = h_{tr}(\boldsymbol{x})$, y sabiendo que al querer maximizar 2.11 el valor de $p(\boldsymbol{x})$ es constante, entonces podemos utilizar en 2.11:

$$p_{tr}(\boldsymbol{x}|y) \equiv h_{tr}(\boldsymbol{x}) / \hat{p}_{tr}(y) \tag{2.13}$$

Un modelo generativo que utiliza un clasificador discriminativo como intermediario para estimar $p(\boldsymbol{x}|y)$ a partir de $p(y|\boldsymbol{x})$ (es decir, el método *LR-Implicit*) pertenece en realidad a los métodos agregativos (mencionados en la Sección 2.1). No obstante, dado que el marco generativo presentado por Keith and O'Connor [39] solo requiere un modelo condicionado por las etiquetas de clase, como ocurre con las versiones explícitas (usando los modelos *MNB y Loglin* como ejemplo), enmarcamos este método más general dentro de los métodos no agregativos.

Ejemplo: En este caso utilizaremos el método *LR-Implicit* aplicado al clasificador con el que venimos trabajando. Utilizaremos el método de grilla para buscar el máximo de la curva de $MLL_{tst}(\theta)$:

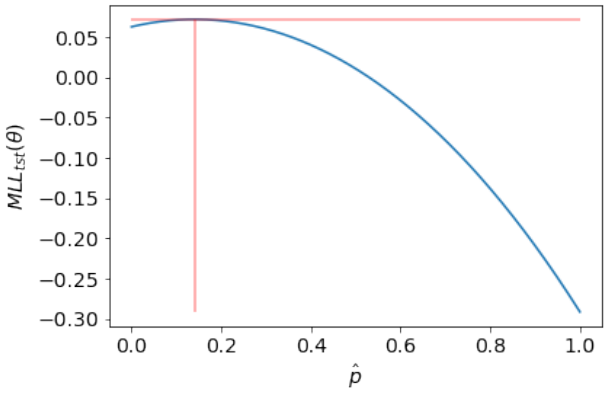


Figura 2.5

Se observa que el valor mínimo de *LR-Implicit* se da en $p_{tst}^{LR-Implicit}(c = 1) = 0.14$.

Capítulo 3

Métodos de Evaluación

La evaluación de métodos de cuantificación es más compleja que en otros problemas. En aprendizaje supervisado, típicamente se mide el rendimiento estimando la probabilidad de predecir correctamente ejemplos individuales no observados. Sin embargo, en cuantificación, el rendimiento se evalúa para conjuntos de datos. Esto implica que necesitamos una colección de muestras para evaluar el rendimiento de un método. Dado un método $\overline{h}$, una función de pérdida $L(\cdot,\cdot)$, y un conjunto de muestras de evaluación $T_1,\ldots,T_s$, el rendimiento de $\overline{h}$ es:

$$Rendimiento(\overline{h}, L, T_1, \ldots, T_s) = \frac{1}{s}\sum_{j=1}^{s} L(\overline{h}, T_j) \tag{3.1}$$

Calcular la pérdida de un método sobre una sola muestra de prueba, $L(\overline{h}, T_j)$ no implica promediar la pérdida sobre ejemplos individuales. Las métricas empleadas para evaluar la cuantificación con respecto a los ejemplos individuales son inherentemente no lineales. Esto implica que el error asociado a un conjunto de elementos no etiquetados no puede ser expresado como una combinación lineal de los errores individuales. Esta no linealidad surge debido a la interdependencia entre la clasificación de los diferentes elementos. Por ejemplo, para el caso binario, si en una muestra no etiquetada hay una mayor incidencia de falsos positivos que de falsos negativos, la presencia de un falso negativo adicional puede en realidad mejorar el error de cuantificación general, gracias al efecto de compensación mutua entre FP y FN mencionado en 1.5. Por consiguiente, la evaluación del error en la cuantificación es de naturaleza no lineal y múltiple, requiriendo así la consideración conjunta de todos los elementos no etiquetados a la vez.

Además, el problema de evaluación en cuantificación se relaciona con el cambio en la distribución de datos entre la fase de entrenamiento y la de implementación del método. Se requiere una colección de muestras de prueba variada y que represente diversas distribuciones para evaluar correctamente el rendimiento del método y evitar sesgos. Por esta razón, la mayoría de los experimentos reportados en la literatura emplean conjuntos de datos tomados de otros problemas y se crean conjuntos de prueba con cambios en las distribuciones creados artificialmente. Este enfoque tiene la ventaja de que la cantidad del *dataset shift* se puede controlar para estudiar el rendimiento de los métodos en diferentes situaciones.

Las funciones de pérdida $L(\cdot,\cdot)$ serán elegidas de acuerdo al tipo de problema y al objetivo particular de la aplicación. Como ya se mencionó, el rendimiento de $\overline{h}$ será el promedio del resultado de la función de pérdida por cada muestra de evaluación, de acuerdo a la ecuación 3.1.

Se han propuesto en la literatura distintas métricas de evaluación para problemas de *Single-Label Quantification (SLQ)*. Estas también se pueden usar para *Binary Quantification (BQ)*, ya que es un caso espacial del anterior, y para *Multi-Label Quantification*, ya que se pueden usar para cada $y \in C$. Esencialmente todas las medidas de evaluación que se han propuesto son divergencias, es decir, medidas de cómo una distribución difiere de otra. No se desarrollarán en esta tesis métricas para *Ordinal Quantification* ni para *Regression Quantification*, ya que no son útiles para nuestro objeto de estudio.

3.1. Propiedades

Sebastiani [40] define una serie de propiedades interesantes para medidas de evaluación en *SLQ*. Un importante resultado de este artículo es que ninguna medida de evaluación existente para *SLQ* satisface todas las propiedades identificadas como deseables; aún así, se ha demostrado que algunas medidas de evaluación son "menos inadecuadas" que otras. Aquí mencionamos brevemente las cuatro propiedades principales que habría que considerar en cada métrica M a emplear (el resto son propiedades que suelen ser satisfechas por todas las métricas).

- **Máximo (MAX)**: si $\exists \beta > 0, \beta \in \mathbb{R}$ tal que por cada $c \in C$ y por cada p, (i) existe $\hat{p}$ tal que $M(p, \hat{p}) = \beta$, y (ii) para ninguna $\hat{p}$ se cumple que $M(p, \hat{p}) > \beta$. Si se cumple **MAX**, la imagen de M es independiente del problema, y esto permite juzgar si un valor dado significa un error de cuantificación alto o bajo. Si M no cumple cumple **MAX**, cada muestra de evaluación tendrá un peso distinto en el resultado final.

- **Imparcial (IMP)**: si M penaliza igualmente la subestimación de p por una cantidad a (es decir, con $\hat{p} = p - a$) o su sobreestimación por la misma cantidad a (es decir, con $\hat{p} = p + a$). Si se cumple **IMP**, la subestimación y la sobreestimación se consideran igualmente indeseables. Esto es generalmente lo deseable, a menos que exista una razón específica para no hacerlo.

- **Relativo (REL)**: si M penaliza más gravemente un error de magnitud absoluta a (es decir, cuando $\hat{p} = p \pm a$) si p es menor. Por ejemplo, predecir $\hat{p} = 0.0101$ cuando $p = 0.0001$ es un error mucho más serio que predecir $\hat{p} = 0.1100$ cuando $p = 0.1000$.

- **Absoluto (ABS)**: si M penaliza un error de magnitud independientemente del valor de p. Mientras algunas aplicaciones requieren **REL**, otras requieren **ABS**. Si bien **REL** y **ABS** son mutuamente excluyentes, ninguna cubre el caso cuando M considera un error de magnitud absoluta a menos grave cuando p es menor (como en el caso de la *distancia coseno*).

3.2. Métricas

Sesgo

El sesgo o *bias* técnicamente no es una medida de evaluación para la cuantificación, ya que no se aplica a toda una distribución p sino solo a una clase específica $c \in C$, y se define como:

$$\mathrm{B}(c) = \hat{p}(c) - p(c) \tag{3.2}$$

Incluso usado en cuantificación binaria, se debe especificar a cuál de las clases hace referencia (en este caso, suele hacer referencia a la clase positiva). Si se usa como una medida de evaluación para la cuantificación, un problema con B es que promediar los puntajes de diferentes clases produce resultados poco intuitivos, ya que el sesgo positivo de una clase y el sesgo negativo de otra clase se anulan entre sí. El mismo problema ocurre cuando se trata de la misma clase pero se promedia entre diferentes muestras. Como resultado, esta medida se puede utilizar como mucho para determinar si un método tiene una tendencia a subestimar o sobrestimar la prevalencia de una clase específica (típicamente la clase minoritaria) en BQ, y no como una medida de evaluación general para usar.

Error Absoluto

El error absoluto o *absolute error* es una de las medidas más empleadas ya que, al ser simplemente la diferencia entre ambas magnitudes, es simple y fácilmente interpretable.

$$\mathrm{AE}(p, \hat{p}) = \frac{1}{\#C} \sum_{j=1}^{\#C} |\hat{p}(c = c_j) - p(c = c_j)| \tag{3.3}$$

Como en este caso las diferencias positivas y negativas son igualmente indeseables, promediar el AE entre varias clases, o varias muestras, no es problemático. Como se muestra en [40], AE cumple **IMP** y **ABS** pero no cumple **MAX** (ni tampoco **REL**). Su rango va de 0 (mejor) a:

$$z_{\mathrm{AE}} = \frac{2(1 - \min\limits_{j \in \{1,\ldots,\#C\}} p(c = c_j))}{\#C} \tag{3.4}$$

(peor), por lo que su rango depende de la distribución de p y de $\#C$.

Error Absoluto Normalizado

El error absoluto normalizado *normalised absolute error*, definido como:

$$\mathrm{NAE}(p, \hat{p}) = \frac{\mathrm{AE}(p, \hat{p})}{z_{\mathrm{AE}}} = \frac{\sum\limits_{j=1}^{\#C} |\hat{p}(c = c_j) - p(c = c_j)|}{2(1 - \min\limits_{j \in \{1,\ldots,\#C\}} p(c = c_j))} \tag{3.5}$$

es una versión de AE que oscila entre 0 (mejor) y 1 (peor), por lo que cumple **MAX**. A pesar de su nombre, NAE no disfruta de **ABS** (ni tampoco **REL**).

Error Cuadrático

El error cuadrático o *squared error*, definido como:

$$\mathrm{SE}(p, \hat{p}) = \frac{1}{\#C} \sum_{j=1}^{\#C} (\hat{p}(c = c_j) - p(c = c_j))^2 \tag{3.6}$$

comparte los mismos pros y contras de AE, pero penalizando más cuanto mayor es la diferencia entre el valor real y el predicho, por lo que se usa cuando se quiere castigar los valores atípicos u *outliers*.

Error Absoluto Relativo

El error absoluto relativo o *relative absolute error* es una adaptación del AE que impone **REL** al hacer que AE sea relativo a p.

$$\mathrm{RAE}(p, \hat{p}) = \frac{1}{\#C} \sum_{j=1}^{\#C} \frac{|\hat{p}(c = c_j) - p(c = c_j)|}{p(c = c_j)} \tag{3.7}$$

RAE cumple **IMP** y **REL** pero no cumple **MAX** (ni **ABS**, a pesar de su nombre). Su rango va de 0 (mejor) a:

$$z_{\mathrm{RAE}} = \frac{\#C - 1 + \frac{1 - \min\limits_{j \in \{1,\dots,\#C\}} p(c = c_j)}{\min\limits_{j \in \{1,\dots,\#C\}} p(c = c_j)}}{\#C} \tag{3.8}$$

(peor), por lo que su rango depende de la distribución de p y de $\#C$.

Error Absoluto Relativo Normalizado

El error absoluto relativo normalizado *normalised relative absolute error*, definido como:

$$\mathrm{NRAE}(p, \hat{p}) = \frac{\mathrm{RAE}(p, \hat{p})}{z_{\mathrm{RAE}}} = \frac{\sum\limits_{j=1}^{\#C} \frac{|\hat{p}(c=c_j) - p(c=c_j)|}{p(c=c_j)}}{\#C - 1 + \frac{1 - \min\limits_{j \in \{1,\dots,\#C\}} p(c = c_j)}{\min\limits_{j \in \{1,\dots,\#C\}} p(c = c_j)}} \tag{3.9}$$

es una versión de RAE que oscila entre 0 (mejor) y 1 (peor), por lo que cumple **MAX**. A pesar de su nombre, NRAE no disfruta de **REL** (ni tampoco **ABS**).

Tanto RAE como NRAE no están definidas cuando sus denominadores sean nulos. Para resolver este problema, se puede suavizar tanto $p(c = c_j)$ como $\hat{p}(c = c_j)$ mediante suavizado aditivo:

$$\underline{p}(c = c_j) = \frac{\epsilon + p(c = c_j)}{\epsilon \#C + \sum\limits_{j=1}^{\#C} p(c = c_j)} \tag{3.10}$$

donde $\underline{p}(c = c_j)$ es la versión suavizada de $p(c = c_j)$ y el denominador es solo un un factor de normalización (lo mismo para $\underline{\hat{p}}(c = c_j)$).

Divergencia de Kullback-Leibler

Para distribuciones de probabilidad discretas P y Q definidas en el mismo espacio muestral $\mathcal{X}$ su divergencia KL se define como:

$$\mathrm{DKL}(P \parallel Q) = \sum_{x \in X} P(x) \log \left(\frac{P(x)}{Q(x)} \right) \tag{3.11}$$

En cuantificación, se quiere comparar la prevalencia real p y la prevalencia predicha $\hat{p}$, y el espacio muestral corresponde a las posibles clases, con lo cuál será:

$$\mathrm{DKL}(p \parallel \hat{p}) = \sum_{j=1}^{\#C} p(c = c_j) \log \left(\frac{p(c = c_j)}{\hat{p}(c = c_j)} \right) \tag{3.12}$$

que va de 0 (mejor) a $+\infty$ (peor) -por lo tanto, no cumple con **MAX**-. Si bien esta medida es una distancia, no es una métrica verdadera, ya que no obedece a la desigualdad del triángulo y no es simétrica. Además, es menos interpretable que otras métricas de rendimiento y no está definido cuando $\hat{p}$ es 0 o 1.

Divergencia de Kullback-Leibler Normalizada

Para suplir los problemas de DKL, se puede utilizar la función logística, quedando:

$$\mathrm{NDKL}(p \parallel \hat{p}) = 2 \cdot \frac{e^{\mathrm{DKL}(p\|\hat{p})}}{1 + e^{\mathrm{DKL}(p\|\hat{p})}} - 1 \tag{3.13}$$

que también va de 0 (mejor) a $+\infty$ (peor) -por lo tanto, si cumple con **MAX**-. Sin embargo, como se muestra en [40], ni DKL ni NDKL cumplen con **IMP**, **REL** y **ABS**, lo que hace que su uso como medidas de evaluación para cuantificación sea cuestionable, además de ser difíciles de interpretar.

3.3. Elección de la Métrica

Es evidente que ninguna de las medidas propuestas hasta ahora es completamente satisfactoria. DKL y NDKL son los menos satisfactorios y parecen fuera de discusión. Respecto a los demás, el problema es que **MAX** parece ser incompatible con **REL**/**ABS**, y viceversa.

Sebastiani [40] sostiene que cumplir con **REL** o **ABS** parece más importante que cumplir con **MAX**, ya que reflejan las necesidades de la aplicación; si no se satisfacen estas propiedades, se puede argumentar que el error de cuantificación que se está midiendo está vagamente relacionado a lo que el usuario realmente quiere. Si **MAX** no está satisfecho, los resultados obtenidos en muestras caracterizadas por diferentes distribuciones no serán comparables. A pesar de esto, los resultados obtenidos por diferentes sistemas en el mismo conjunto de muestras siguen siendo comparables.

Esto sugiere que AE, RAE y SE son las mejores medidas a elegir. Se debe preferir AE cuando un error de estimación de una magnitud absoluta dada debe considerarse más grave cuando la verdadera prevalencia de la clase afectada es menor. RAE debe ser elegido cuando un error de estimación de una magnitud absoluta dada tiene el mismo impacto independientemente de la verdadera prevalencia de la clase afectada. Si se quiere penalizar mayormente errores atípicos, considerando mucho más graves a los errores cuanto mayor es la diferencia entre el valor real y el predicho, entonces SE es la métrica más conveniente.

3.4. Protocolos

Mientras que en la clasificación, un conjunto de datos de tamaño k proporciona k puntos de evaluación, para la cuantificación, el mismo conjunto solo proporciona 1 punto. Evaluar algoritmos de cuantificación es por lo tanto un reto, debido a que la disponibilidad de datos etiquetados con fines de prueba es más restringido. Hay principalmente dos protocolos experimentales que se han tomado para tratar con este problema: el Protocolo de Prevalencia Natural (*NPP*) y el Protocolo de Prevalencia Artificial (*APP*).

- *NPP*: Consiste en, una vez entrenado un cuantificador, tomar un conjunto de prueba (no observado en el entrenamiento) lo suficientemente grande, dividirlo en un número de muestras

de manera uniformemente aleatoria, y llevar a cabo la evaluación individualmente en cada muestra.

- *APP*: Consiste en, previo al entrenamiento, tomar un conjunto de datos, dividirlo en un conjunto de entrenamiento y en un conjunto de evaluación de manera aleatoria, y realizar experimentos repetidos en los que la prevalencia del conjunto de entrenamiento o la prevalencia del conjunto de prueba de una clase se varía artificialmente a través del submuestreo.

Ambos protocolos tienen diferentes pros y contras. Una ventaja de *APP* es que permite crear muchas puntos de prueba de la misma muestra. Además, *APP* permite simular distintos *Prior probability shift*, mientras que con *NPP* se estaría evaluando sólo con las distribuciones originales de los datos de entrenamiento y prueba. Sin embargo, una desventaja de *APP* es que puede no saberse cuán realistas son estas diferentes situaciones en la aplicación real, por lo que se podría estar destinando recursos a una evaluación errónea o pobre. Una solución intermedia podría ser utilizar un protocolo que utilice conocimientos previos sobre la distribución de prevalencias "probables" que se podría esperar encontrar en el dominio específico en cuestión.

3.5. Selección de Modelos

El rendimiento de muchos algoritmos de aprendizaje automático es altamente sensible a la configuración de sus hiperparámetros. Estos hiperparámetros, a diferencia de los parámetros, no se aprenden durante el entrenamiento, debiendo ser ajustados previamente para cada problema específico. Aunque muchos algoritmos ofrecen valores predeterminados, la optimización de estos hiperparámetros es esencial para maximizar el rendimiento en aplicaciones concretas. Los métodos de cuantificación no son una excepción en este sentido [16].

El proceso de selección de modelos consiste en evaluar diferentes combinaciones de hiperparámetros hasta quedarse con la combinación que obtenga la mejor métrica. Para garantizar una evaluación rigurosa, es recomendable utilizar validación cruzada. En el caso de la cuantificación, la optimización de hiperparámetros debería imitar el protocolo de evaluación al evaluar cada una de las configuraciones candidatas. En otras palabras, dado que el objetivo de la selección de modelos es encontrar la configuración de hiperparámetros que funcione mejor de acuerdo con un protocolo experimental dado y una medida de evaluación dada, resulta adecuado adoptar los mismos protocolos de evaluación (3.4) y métricas (3.2) que se usan habitualmente en la evaluación de sistemas de cuantificación [41, 42].

Algunos de los métodos de cuantificación que presentan hiperparámetros *per-se* y que hemos mencionado en el Capítulo 2 son el *HDy* y *HDx* (con respecto a la cantidad de *bins*) y *ACC* y *PACC* para el caso multiclase (con respecto al método para resolver el sistema de ecuaciones lineales), entre otros. Adicionalmente, todos los métodos agregativos (2.1) heredan los hiperparámetros de los clasificadores que emplean. En este sentido, Moreo y Sebastiani [36] afirman que el método *CC* y sus variantes, con una adecuada optimización, pueden ofrecer un rendimiento competitivo, aunque siguen siendo inferiores a los métodos de cuantificación más sofisticados, además de requerir recursos y tiempo de cómputo para la búsqueda de hiperparámetros.

Capítulo 4

Experimentos

En esta sección se evaluarán todos los métodos mencionados en el Capítulo 2 mediante nuevos casos de ejemplo simulados. Dicha evaluación se basará en las métricas elegidas en 3.3 (RAE, AE y SE) además del sesgo. Además, la simulación se basará en el protocolo *APP* definido en 3.4. No se realizará el proceso de selección de modelos mencionado en 3.5 ya que el objetivo de estos experimentos no es el de buscar los mejores cuantificadores para los casos de ejemplo, sino hacer una comparación de los mismos frente a iguales condiciones (hiperparámetros por defecto, mismo clasificador, etc.). Finalmente, se elaboran conclusiones en base a los resultados y se comparan con los resultados de otros trabajos [36, 41, 43–45].

4.1. Simulación

4.1.1. Poblaciones

La simulación consiste, en primer lugar, en generar dos poblaciones sintéticas de datos. Al igual que en 2, para sintetizar los datos se utilizó el algoritmo propuesto por Guyon [23] mediante la función *make_classification* de *scikit-learn*. En este caso, se crearon dos poblaciones, F (fácil) y D (difícil), cuyos argumentos de creación son (los parámetros no mencionados usan sus valores por defecto):

	Población	
	F	D
n_samples	51000	51000
n_features	2	100
n_informative	2	5
n_redundant	0	15
n_repeated	0	15
flip_y	0	0
class_sep	0.6	0.2

Tabla 4.1: Poblaciones

La elección de estos valores tiene como objetivo generar una población fácil de clasificar y con características de baja dimensionalidad, y otra más difícil y con alta dimensionalidad, para probar

bajo estas dos condiciones cada método. Además, al usar los parámetros *n_classes* y *weights* por defecto, estaremos generando datasets binarios y perfectamente balanceados.

4.1.2. Datasets

Tenemos entonces dos poblaciones sintéticas, cada una de 51000 individuos. Luego, procedemos a sub-dividir ambas poblaciones de forma estratificada (es decir, manteniendo la proporción de clases balanceadas) para crear los datasets de entrenamiento y de prueba en cada población. Dichos datasets son de 50000 y 1000 individuos respectivamente.

Siguiendo el protocolo *APP* ya mencionado, la simulación consistió en muestrear de forma iterativa el dataset de entrenamiento de cada población con distintos tamaños de muestra (500 y 5000), distintas prevalencias[1] (0.01, 0.25, 0.5, 0.75 y 0.99) y de forma repetida (5 veces) -es decir, un total de 50 muestras distintas-. Luego, por cada muestra de entrenamiento, a su vez, se muestreo de forma iterativa el dataset de evaluación. En este caso también se tomaron distintos tamaños de muestra (10 y 100), distintas prevalencias (0, 0.2, 0.5, 0.8 y 1.0 para las muestras de tamaño n=10, y 0.01, 0.25, 0.5, 0.75 y 0.99 para las muestras de tamaño n=100) y de forma repetida (5 veces) -50 muestras de prueba por cada muestra de entrenamiento-.

	Dataset			
	Train		Test	
n_samples	500	5000	10	100
prev	0.01		0	0.01
	0.25		0.2	0.25
	0.5		0.5	0.5
	0.75		0.8	0.75
	0.99		1.0	0.99
n_repetitions	5		5	

Tabla 4.2: Datasets

4.1.3. Cuantificación

Por cada muestra de entrenamiento, se procede a ajustar cada uno de los métodos a evaluar. Empezando por los métodos agregativos que utilizan clasificadores generales (2.1.1), tenemos los que requieren como entrada las etiquetas de las clases predichas (es decir, que usan clasificadores duros). En nuestro caso, estos métodos son el *CC* y el *ACC*. Es decir, que primero debemos ajustar un modelo de clasificación. En esta simulación hemos utilizado dos modelos distintos, un modelo de regresión logística y el clasificador de XGBoost, con la idea de probar un modelo simple y uno complejo, con el objetivo de determinar si la complejidad del modelo mejora o no la cuantificación. En este caso no es necesario calibrarlos ya que usaremos solamente las etiquetas predichas. Para sus hiperparámetros usaremos los valores por defecto de las librerías *scikit-learn* y *xgboost* respectivamente.

[1]Para evitar repeticiones, cuando mencionemos prevalencia en estos casos se hará referencia siempre a la clase positiva.

Clasificación	
Modelo	Complejidad
LogisticRegression	Baja
XGBoost	Alta

Tabla 4.3: Modelos de clasificación

En cambio, para los métodos agregativos que utilizan clasificadores generales pero que requieren como entrada las probabilidades *a posteriori* de pertenencia a cada clase (es decir, clasificadores blandos), debemos no solo ajustar los modelos de clasificación, sino que también conviene calibrarlos (ya que es un supuesto de estos métodos). En nuestro caso, estos métodos son el *PCC*, *PACC*, *EMQ* y *HDy*. Con respecto a los algoritmos de calibración, utilizaremos los cuatro métodos para modelos binarios propuestos por Guo et al. [46] y mencionados en el Apéndice A, además de probar también los clasificadores sin calibrar. Como los métodos de calibración requieren utilizar validación cruzada para ajustar sus parámetros, utilizaremos el método de *held-out*, es decir que destinaremos una porción (20 %) del dataset de entrenamiento para ello.

Calibración	
Método	Held-out
No Calibration	0 %
Histogram Binning	20 %
Isotonic Regression	20 %
BBQ	20 %
Platt Scaling	20 %

Tabla 4.4: Modelos de calibración

Para los métodos agregativos que utilizan clasificadores específicos (2.1.2) -en nuestro caso, únicamente el método *ELM*-, debemos también ajustar un modelo de clasificación previo a la cuantificación, pero en este caso no utilizaremos los nombrados en 4.3, sino uno optimizado para ser utilizado en cuantificación. En este caso usaremos, al igual que en el Capítulo 3, el clasificador SVM_{perf}, pero esta vez optimizado no solo para la métrica KLD, sino también AE y RAE.

Para los modelos no agregativos (2.2), no se usan modelos de clasificación, y por lo tanto tampoco se calibran; simplemente se aplica directamente el método de cuantificación. En nuestro caso, para este grupo usaremos solo el método *HDx*, ya que no usaremos modelos de clasificación generativos explícitos. Sin embargo, sí usaremos el método *LR-Implicit*, el cual, como bien dijimos, se debe considerar en realidad un método agregativo, y así lo haremos en las simulaciones.

Cabe mencionar que para los métodos de cuantificación que requieren de validación cruzada para estimar sus parámetros internos también utilizaremos el método de *held-out*, destinando otra porción (20 %) del dataset de entrenamiento para ello.

Cuantificación	
Método	Held-out
CC	0 %
ACC	20 %
PCC	0 %
PACC	20 %
TH_MAX	20 %
TH_X	20 %
TH_T50	20 %
TH_MS	20 %
EMQ	0 %
HDy	20 %
HDx	0 %
ELM-SVMperfKLD	0 %
ELM-SVMperfAE	0 %
ELM-SVMperfRAE	0 %
LR-Implicit	0 %

Tabla 4.5: Modelos de cuantificación

Finalmente, una vez ajustado el método de cuantificación para una muestra de entrenamiento, se procede a tomar una muestra de prueba (recordando que probaremos con distintos tamaños de muestra, prevalencia y repitiendo el muestreo) y estimar su prevalencia.

Para la implementación de los métodos de cuantificación en la simulación se utilizó la librería *Quapy* [43] (`https://github.com/HLT-ISTI/QuaPy`), excepto para el método *LR-Implicit* [39], para el cual se empleó el código provisto por los autores (`https://github.com/slanglab/freq-e`). Para la implementación de los modelos de calibración utilizamos la librería *net:cal* [47] (`https://github.com/EFS-OpenSource/calibration-framework`).

4.1.4. Evaluación

Como ya se mencionó, se realizó la evaluación en base a las métricas elegidas en 3.3, es decir, se utilizaron RAE, AE y SE. La evaluación se hizo según distintos criterios, agrupando los resultados en base a estos criterios y calculando su intervalo de confianza del 95 %. Además, en los casos que corresponda, usaremos también la métrica de clasificación F1 y la métrica de calibración ECE para elaborar conclusiones.

4.2. Conclusiones

Empezamos por analizar el rendimiento de los cuantificadores en general. En la tabla B.1 se muestran los resultados de las estimaciones de las métricas RAE, AE y SE para cada cuantificador usado en la simulación. Los métodos basados en selección de umbrales (*TH*), *PACC*, *LR-Implicit* y *EMQ* son los que en general mejor desempeño tienen, mientras que los basados en minimización de pérdida explícita (*ELM*) son los peores. Estos resultados son congruentes con los de Schumacher et al. [45] (en donde *TH_MS* es el método con mejor AE), los de Moreo and Sebastiani [36] (que

afirma que *PACC* es el mejor método dentro de las variantes de *CC*), los de Moreo et al. [43] (donde *EMQ* resultó el método con mejor AE), los de Moreo and Sebastiani [41] (donde también *EMQ* y *PACC* resultaron los mejores métodos evaluados con AE) y los de Tasche [44] (que concluye que los métodos *CC* y *PCC* son limitados frente a *ACC* y *PAC*).

Si queremos ver si los métodos sobre o sub estiman las prevalencias, debemos entonces analizar el sesgo. Además, en este caso debemos mirar la distribución del mismo, ya que su estimación puntual podría dar baja por más que los métodos tengan error alto (ya que pueden cancelarse entre sí). Podemos observar en la figura 4.1 que *HDy* tiende a subestimar, mientras que el resto de los métodos no presentan, en general, un sesgo marcado. En esta figura también se verifica que *TH_MS* tiene el mejor comportamiento. Sin embargo, si analizamos el sesgo según la verdadera prevalencia en la muestra de prueba (figura 4.2), todos los métodos si tienen un sesgo en favor de la clase minoritaria.

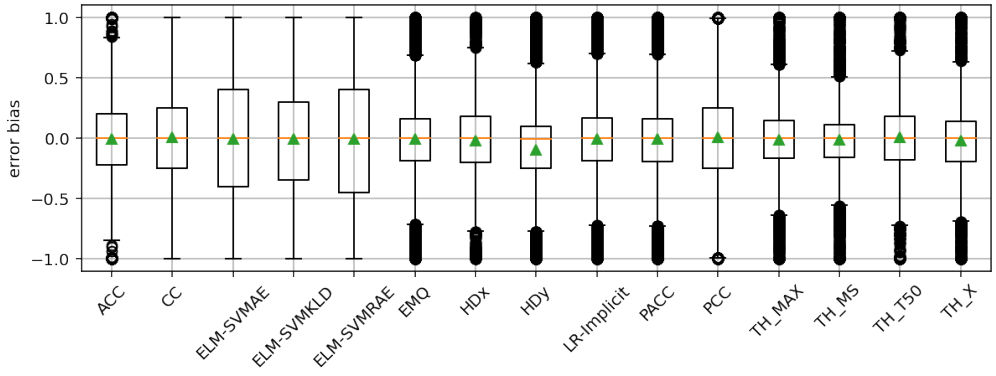


Figura 4.1: Sesgo por método de cuantificación

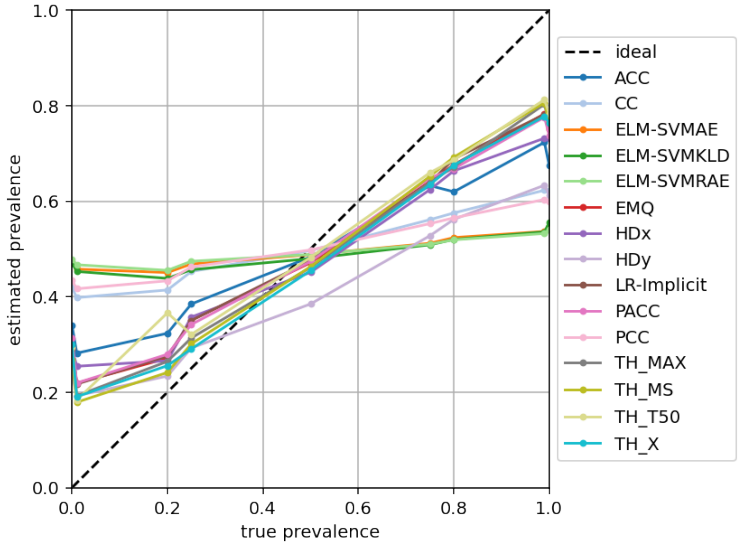


Figura 4.2: Prevalencia estimada por método de cuantificación según prevalencia de prueba

Por otro lado, también verificamos algunas de los resultados que eran de esperarse:

- Para los métodos de cuantificación agregativos con clasificadores generales (2.1.1), a mejor desempeño en la clasificación (métrica F1), mejor desempeño en la cuantificación (tabla B.2).

- A menor cantidad de características y mayor separación entre clases en las poblaciones, mejor rendimiento de cuantificación (tabla B.3).

- A mayor tamaño en muestra de entrenamiento, mejor rendimiento de cuantificación (tabla B.4) (coincide con los resultados de Schumacher et al. [45]).

Si analizamos según el tamaño de la muestra de evaluación (tabla B.5), teniendo que para cada tamaño utilizamos distintas posibles prevalencias como definimos en la tabla 4.2, y recordando que los valores de las métricas dependen de los mínimos valores posibles de las prevalencias de prueba (ecuaciones 3.4 y 3.8), entendemos entonces la diferencia de resultados en las métricas entre los distintos tamaños de muestra.

Vemos también que cuanto más balanceada sea la muestra de entrenamiento, mejores resultados de cuantificación se obtienen (tabla B.6 y figuras 4.3 y 4.4). Este es un dato bastante interesante, ya que implicaría que podríamos aplicar en cuantificación las mismas técnicas de balance de clases que se usan en problemas de clasificación para tratar de conseguir mejores resultados que si usáramos datos desbalanceados. También vemos como la prevalencia de la muestra de entrenamiento influye en el sesgo del cuantificador, tendiendo en general a sobrestimar la clase mayoritaria. En la figura 4.5 podemos ver estos resultados desagregados por método de cuantificación.

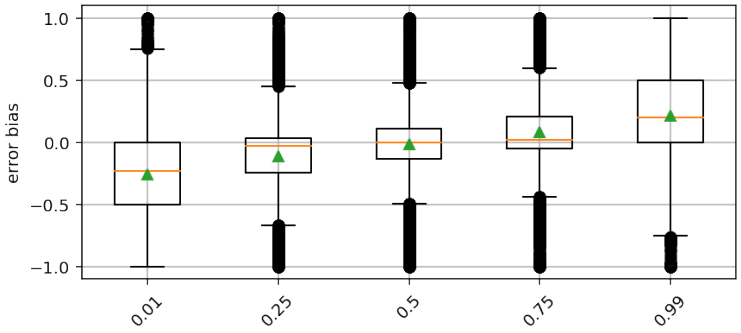


Figura 4.3: Sesgo por prevalencia de entrenamiento

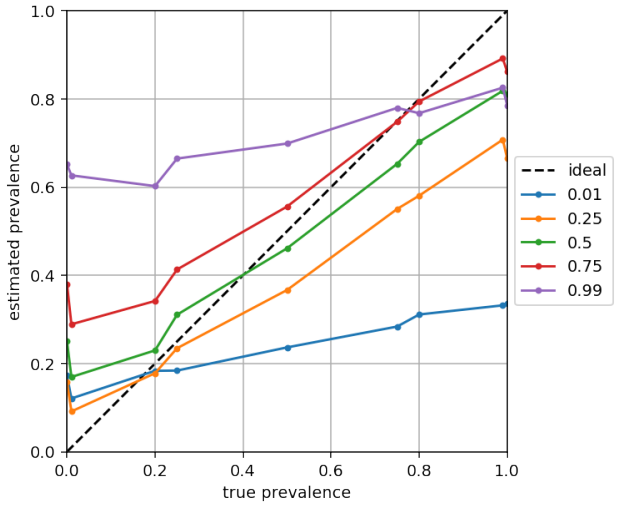


Figura 4.4: Prevalencia estimada por prevalencia de entrenamiento según prevalencia de prueba

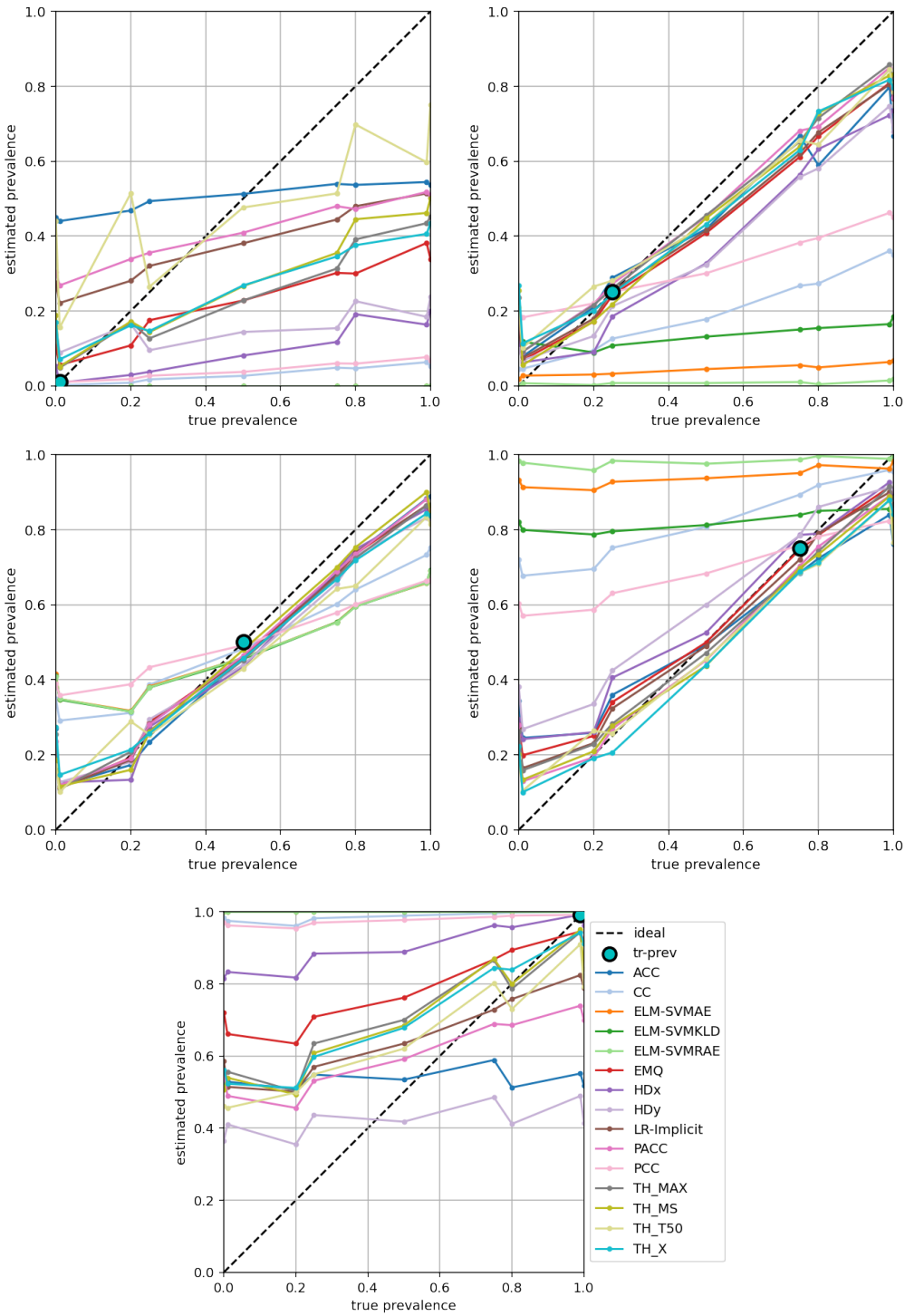


Figura 4.5: Prevalencia estimada por método de cuantificación según prevalencia de prueba, para distintas prevalencias de entrenamiento

Con respecto al balance en la muestra de prueba (tabla B.7), debemos volver a mencionar el comentario en referencia a los distintos valores de prevalencia para los distintos posibles tamaños de muestra. De todas formas, llama la atención que las métricas son mejores cuando el desbalance de clases lleva a la clase positiva a ser la minoritaria. Esto se debe a que los métodos de cuantificación binaria basados en la estimación del *fpr* y *tpr* (*ACC*, *PACC*, y *TH*) no son simétricos en cuanto a las clases positivas y negativas, sino que son sensibles a si la clase mayoritaria es una u otra.

Si tenemos en cuenta la diferencia absoluta entre las prevalencias de la muestra de entrenamiento y la de prueba (tabla B.8), vemos que también en general (y como quizás era esperable de forma intuitiva), a menor diferencia, mejor rendimiento. Sin embargo, llama la atención cuan dinámico es este comportamiento según el método de cuantificación (tabla B.9). En general, los métodos que mejor funcionan cuando el *dataset shift* es bajo son los que peor funcionan cuando es alto. A pesar de que en Moreo and Sebastiani [41] y Moreo et al. [43] el método *EMQ* es el de mejor rendimiento para los casos de mayor *dataset shift*, en nuestras simulaciones fue *TH_T50* el método con mejores métricas ante estos casos.

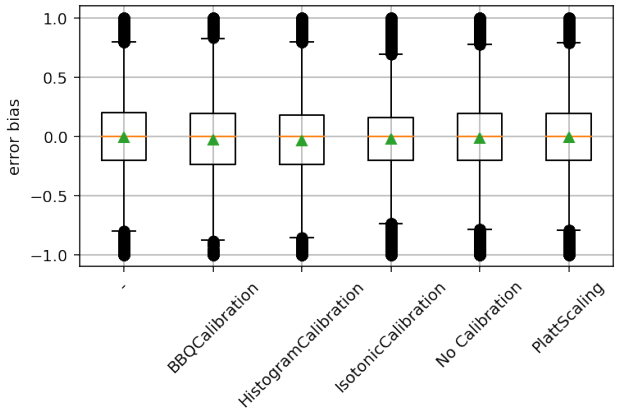


Figura 4.6: Sesgo por método de calibración

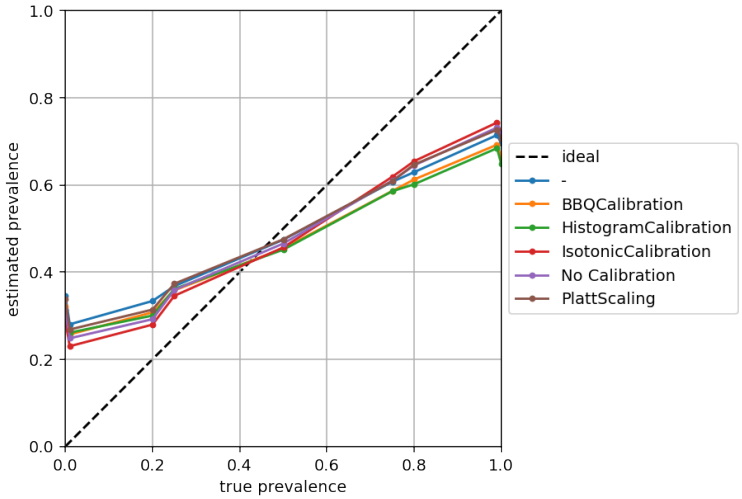


Figura 4.7: Prevalencia estimada por método de calibración según prevalencia en muestra de prueba

Analizando ahora la influencia de la calibración en la cuantificación, la calibración por regresión isotónica lleva una leve ventaja en el rendimiento de la cuantificación en general (considerando solo métodos de cuantificación agregativos con clasificadores generales y blandos) (tabla B.10 y figuras 4.6 y 4.7), aunque llamativamente no presenta ventajas en cuánto a la métrica de clasificación (F1) ni de calibración (ECE).

Si tenemos en cuenta tanto el método de cuantificación como el de calibración (en los casos que aplique) (tabla B.11 y figura 4.8), se observa que *LR-Implicit*, *EMQ* y *PACC* en los casos de calibración isotónica y de no calibración se vuelven más competitivos frente a los métodos *TH*.

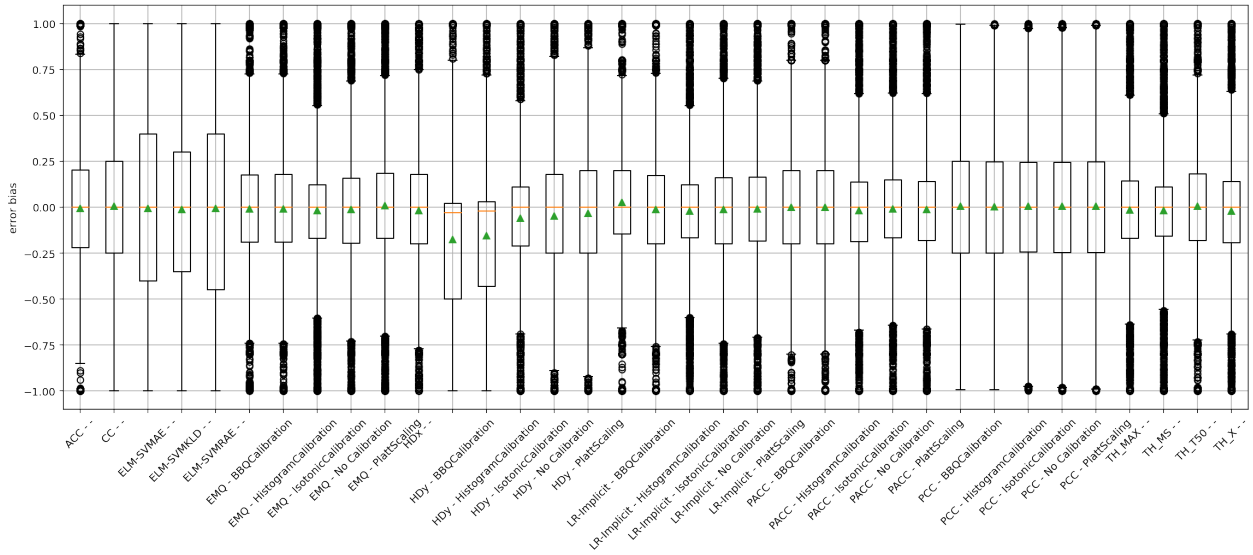


Figura 4.8: Sesgo por método de calibración y cuantificación

Sin embargo, también notamos que esta mejora en la cuantificación mediante la calibración surge cuanto más desbalanceados estén los datos de entrenamiento, no habiendo mejoras significativas cuando están balanceados (tabla B.12).

Apéndice A

Calibración

En problemas de clasificación, el subproblema de la predicción de estimaciones de probabilidad representativas de las probabilidades verdaderas es conocido como calibración. En los sistemas del mundo real, los clasificadores no sólo deben ser precisos, sino que también deben indicar cuando es probable que sean incorrectos. Es decir, deben estimar su nivel de incertidumbre o confiabilidad. En otras palabras, las probabilidades asociadas con la etiquetas de clase predichas deben reflejar su verosimilitud real. Para ejemplificar este concepto traducimos una cita de Nate Silver en su libro *The Signal and the Noise*:

> " Una de las pruebas más importantes del pronóstico del tiempo se llama calibración. De todas las veces que dijiste que había un 40 % de probabilidad de lluvia, ¿con qué frecuencia llovió realmente? Si, a largo plazo, realmente llovió alrededor del 40 % de las veces, eso significa que sus pronósticos estaban bien calibrados. Si terminó lloviendo solo el 20 % de las veces, o el 60 %, no fue así. "

Uno de los casos en donde se debe contemplar este problema es en la toma de decisiones (es decir, en casi toda aplicación real). Por ejemplo, en sistemas utilizados para la salud, un diagnóstico con un bajo nivel de confianza puede significar realizar otro tipo de chequeo. A su vez, las estimaciones de probabilidad, pueden ser utilizadas para ser incorporadas en otro modelo probabilístico. Por ejemplo, se podrían combinar distintas salidas de distintos modelos de forma ponderada para obtener una predicción más robusta frente los casos en donde cada modelo individual falla. Sin embargo, si únicamente es de interés la etiqueta generada por el clasificador, la calibración no aporta valor.

Si bien la calibración parece una propiedad sencilla y quizás trivial, los modelos mal calibrados son bastante comunes. Por ejemplo, algunos modelos complejos están mal calibrados desde su origen, como Naive Bayes, Random Forest y algunas redes neuronales modernas. Incluso, hay modelos que no devuelven probabilidades *a posteriori*, sino genéricas puntuaciones de confianza, como es el caso de SVM. En estos dos últimos casos es posible mapear las salidas de los clasificadores a probabilidades *a posteriori* calibradas a través de algunos método de calibración [46, 48–50].

Dentro de los distintos modelos de clasificación, la regresión logística es un caso especial ya que está bien calibrado por diseño, dado que su función objetivo minimiza la función de pérdida logarítmica o *log-loss* [51]. Sin embargo, si no se cuenta con un conjunto de entrenamiento lo suficientemente grande, es posible que el modelo no tenga suficiente información para calibrar las probabilidades y sufrir la maldición de la dimensionalidad. Las cosas se complican aún más cuando

se incorpora la regularización o *boosting*. Dadas todas las posibles causas de una mala calibración, no se debe asumir que el modelo ajustado estará calibrado.

Por último, una buena calibración no implica que el modelo sea bueno clasificando. Por ejemplo, podemos tener un conjunto de datos balanceado (50 % clase positiva, 50 % clase negativa) con el cual ajustamos un modelo que termina clasificando aleatoriamente las observaciones en cada grupo con un 50 % de probabilidad. Este modelo tendrá solo un 50 % de *accuracy* pero estará calibrado de forma perfecta. Lo mismo puede ocurrir al contrario, un modelo con muy buena capacidad de clasificación pero que siempre sobrestime las probabilidades predichas.

A.1. Definición

Matemáticamente podemos definir la calibración perfecta como:

$$\mathbb{P}(\hat{Y} = y|\hat{P} = p) = p, \forall p \in [0, 1] \tag{A.1}$$

Donde $\hat{Y}$ es la variable aleatoria asociada a la clase predicha y $\hat{P}$ la de la probabilidad *a posteriori* entregada por el modelo de clasificación. La probabilidad $\mathbb{P}$ no puede ser computada de forma práctica ya que $\hat{P}$ es una variable aleatoria continua. Esto motiva la necesidad de aproximaciones empíricas.

A.2. Métodos de Calibración

Se pueden clasificar en binarios o multiclase, y en no paramétricos y paramétricos. Para los métodos no paramétricos, debemos tener acceso a las probabilidades $\hat{p}_i(x_i)$ estimadas por el modelo, y a las y_i clases reales del set de calibración. Para los métodos paramétricos, debemos tener acceso a las salidas no probabilísticas del modelo $z_i(x_i) = logit(\hat{p}_i(x_i))$ o bien calcularlas de a través de las $\hat{p}_i(x_i)$, y a las y_i clases reales del set de calibración.

A.2.1. Modelos Binarios

Métodos no paramétricos

Histograma: Las predicciones $\hat{p}_i(x_i)$ se dividen en *bins* $B_1, \dots B_M$ mutuamente excluyentes. Cada bin se asigna a una probabilidad calibrada θ_m (si $\hat{p}_i(x_i)$ se asigna al bin B_m, entonces $\hat{q}_i = \theta_m$). Al predecir, si la predicción $\hat{p}_t$ cae en el bin B_m, entonces la probabilidad calibrada $\hat{q}_t$ es θ_m. Los límites de los *bins* pueden elegirse para que tengan la misma longitud o bien tengan la misma cantidad de muestras.

Regresión Isotónica: Se estima la función por partes mediante constantes $\hat{q}_i = f(\hat{p}_i)$ que minimiza $\sum_{i=1}^{n}(f(\hat{p}_i(x_i)) - y_i)^2$. Esta técnica es una generalización de la anterior, en la que los límites de los *bins* y las predicciones se optimizan simultáneamente.

Agrupamiento bayesiano en cuantiles (BBQ): Es una extensión del histograma, donde mútiples opciones de *bins* son consideradas y luego combinadas. Todas las opciones consideradas son de igual frecuencia, pero varían en la cantidad de *bins* que emplean. Luego se combinan mediante un score Bayesiano (que no analizaremos por su complejidad).

Métodos paramétricos

Platt Scaling: Las salidas no probabilísticas $z_i(x_i) = logit(\hat{p}_i(x_i))$ del clasificador son usadas como covariables para entrenar un modelo de regresión logística, el cual es entrenado con el set destinado a calibración para que retorne las probabilidades reales. Es decir, la idea es estimar los parámetros $a, b \in IR$ tal que $\hat{q}_i = \sigma(a.z_i + b)$. Estos parámetros se buscan optimizando para la Negative Log-Likelihood.

A.2.2. Modelos Multiclase

Métodos no paramétricos

Una forma de extender los métodos binarios no paramétricos a multiclase es tratando el problema como K problemas de uno-vs-todos. Esto dará K modelos de calibración, cada uno para una clase en particular. Al predecir, obtenemos un vector de probabilidades no normalizado $[\hat{q}_i^{(1)}, \ldots, \hat{q}_i^{(K)}]$ donde $\hat{q}_i^{(k)}$ es la probabilidad calibrada para la clase k. Luego, esto vector es normalizado dividiendo por $\sum_{k=1}^{K} \hat{q}_i^{(k)}$. Esta extensión se puede aplicar a cualquiera de los métodos no paramétricos binarios.

Métodos paramétricos

Vector scaling: Es una extensión del método de Platt Scaling, en donde se aplica una transformación lineal $W.z_i + b$ a los $logits$:

$$\hat{q}_i = \max_k \sigma_{SM}(W.z_i + b)^K$$

Los parámetros W y b se buscan optimizando para la Negative Log-Likelihood. Si se restringe a que W sea diagonal, estamos dentro de la variante Vector Scaling.

Temparature Scaling: Es la extensión más simple de Platt Scaling, ya que usa un sólo parámetro $T > 0$ para todas las clases.

$$\hat{q}_i = \max_k \sigma_{SM}(z_i/T)^K$$

T es llamado temperatura, y suaviza a la función $softmax$ cuando $T > 1$. A medida que $T \to \infty$, $\hat{q}_i \to 1/K$, representando máxima incertidumbre. Con $T = 1$, se recupera la $\hat{p}_i$. A medida que $T \to 0$, $\hat{q}_i \to 1$. El valor de T se busca optimizando con respecto a la Negative Log-Likelihood. Como el parámetro T no altera el máximo de la función $softmax$, la predicción de las clases post-calibración $\hat{y}_i'$ se mantendrán igual a las de antes de calibrar. Es decir, que este método de calibración no cambiará el *accuracy* del modelo.

A.3. Métodos de Evaluación

A.3.1. Diagramas de confiabilidad

También llamadas curvas de calibración, se construyen a partir de los valores verdaderos de las clases y las probabilidades predichas para la clase principal. Para generar una curva de calibración debemos seguir los siguientes pasos:

1. Ordenar las probabilidades predichas por el modelo para la clase principal de menor a mayor.
2. Dividir dichas probabilidades en M bines de tamaño fijo $(1/M)$.
3. Calcular la fracción de verdaderos positivos en cada bin.
4. Calcular el promedio de las probabilidades en cada bin.
5. Graficar la fracción de verdaderos positivos en el eje y el promedio de las probabilidades en el eje x.

Matemáticamente, siendo B_m el set de índices de las muestras cuyas probabilidades predichas caen en el intervalo $I_m = (\frac{m-1}{M}, \frac{m}{M}]$:

$$acc(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} 1(\hat{y}_i = y_i) \tag{A.2}$$

donde $\hat{y}_i$ y y_i son las clases predichas y reales respectivamente. Se puede demostrar que $acc(B_m)$ es un estimador insesgado y consistente de $P(\hat{Y} = Y | \hat{P} \in I_m)$. Por otro lado, tenemos:

$$conf(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i \tag{A.3}$$

donde $\hat{p}_i$ es la probabilidad predicha para la muestra i.

$acc(B_m)$ y $conf(B_m)$ aproximan el lado izquierdo y derecho de A.1 respectivamente, para el bin B_m. Por lo tanto, un modelo perfectamente calibrado tendrá:

$$acc(B_m) = conf(B_m), \forall m \in 1, \ldots, M \tag{A.4}$$

Cuanto mejor calibrado esté el modelo, se aproximará en mayor medida la curva de calibración obtenida a la diagonal. La curva de calibración quedará por encima de la diagonal si el modelo tiende a subestimar las probabilidades y por debajo si las sobreestima.

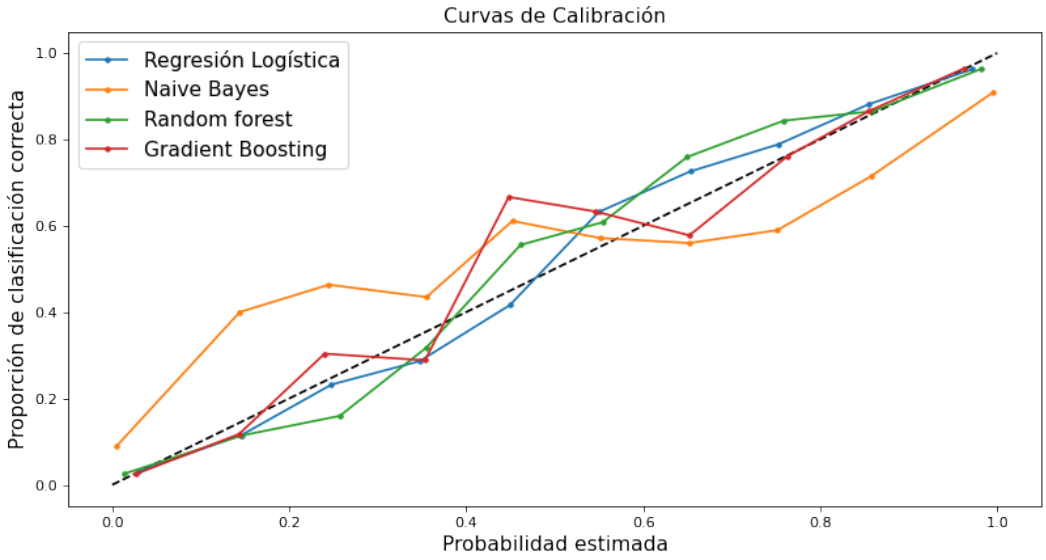


Figura A.1: Ejemplo de curvas de calibración

A.3.2. Métricas

Aunque las curvas de calibración aportan información detallada, es interesante disponer de un único valor que permita cuantificar la calidad de calibración del modelo. Vamos a definir primero dos métricas que también se basan en *bins*, y luego otras que no.

Expected Calibration Error (ECE)

Una posible métrica para medir la descalibración podría basarse en tomar ambos términos A.1, restarlos, y calcular el valor esperado de la diferencia:

$$E_{\hat{P}}[|P(\hat{Y} = Y|\hat{P} = p) - p|]$$

que podemos estimar usando la siguiente fórmula:

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{n} |acc(B_m) - conf(B_m)|$$

donde n es el número de muestras.
El ECE es el promedio de las diferencias de los *bins* en las curvas de calibración.

Maximum Calibration Error (MCE)

Usando el mismo concepto, pero queriendo minimizar la peor desviación, podemos buscar:

$$\max_{p \in [0,1]} |P(\hat{Y} = Y|\hat{P} = p) - p|$$

el cual también se aproxima mediante *bins*:

$$MCE = \max_{p \in [0,1]} |acc(B_m) - conf(B_m)|$$

El MCE es la máxima diferencia de los *bins* en las curvas de calibración.

Cross-Entropy

Teniendo en cuenta que la descalibración puede deberse a un sobreajuste de la Cross-Entropy, podemos medir dicha métrica en el set destinado a calibración.

$$CE = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} \log \hat{p}_{ij}(x_i)$$

Donde N es la cantidad de muestras y M la cantidad de clases. Si este valor es relativamente menor (o mayor) al del entrenamiento, es un síntoma de una posible des-calibración. Un problema de esta métrica es que da infinito si alguna $\hat{p}_{ij} = 0$.

Brier score

Otra posible métrica para medir la bondad de ajuste de las probabilidades es el Brier score, el cual se calcula como:

$$BS = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M}(\hat{p}_{ij}(x_i) - y_{ij})^2$$

Con respecto al Cross-Entropy, tiene la ventaja de que es siempre finito.

Apéndice B

Tablas de Resultados

	RAE			AE			SE		
	media	(	)	media	(	)	media	(	)
Cuantificador									
TH_MS	2.102	1.995	2.208	0.224	0.219	0.229	0.120	0.115	0.124
TH_T50	2.172	2.065	2.279	0.251	0.246	0.257	0.140	0.135	0.144
TH_MAX	2.186	2.085	2.287	0.232	0.227	0.237	0.118	0.114	0.122
TH_X	2.305	2.193	2.418	0.245	0.240	0.250	0.133	0.128	0.138
LR-Implicit	2.361	2.314	2.408	0.239	0.237	0.241	0.122	0.120	0.124
EMQ	2.382	2.331	2.433	0.254	0.252	0.257	0.142	0.140	0.144
PACC	2.399	2.353	2.445	0.231	0.229	0.234	0.112	0.110	0.114
HDx	2.716	2.544	2.888	0.272	0.264	0.279	0.146	0.140	0.153
ACC	2.887	2.775	2.999	0.255	0.250	0.260	0.125	0.121	0.129
HDy	2.958	2.899	3.018	0.302	0.299	0.304	0.185	0.183	0.188
CC	3.909	3.757	4.061	0.362	0.356	0.368	0.228	0.222	0.234
PCC	3.987	3.923	4.051	0.341	0.338	0.343	0.196	0.194	0.198
ELM-SVMKLD	4.569	4.335	4.803	0.407	0.398	0.416	0.271	0.262	0.280
ELM-SVMAE	4.638	4.394	4.882	0.428	0.419	0.437	0.296	0.286	0.305
ELM-SVMRAE	4.724	4.473	4.975	0.440	0.430	0.449	0.312	0.302	0.321

Tabla B.1: Por método de cuantificación

	RAE			AE			SE			F1		
	media	(	)	media	(	)	media	(	)	media	(	)
Clasificador												
XGBClassifier	2.308	2.281	2.336	0.232	0.231	0.233	0.12	0.119	0.121	0.517	0.516	0.519
LogisticRegression	3.240	3.207	3.273	0.310	0.309	0.312	0.18	0.179	0.181	0.384	0.382	0.386

Tabla B.2: Por modelo de clasificación (considerando solo métodos de cuantificación agregativos con clasificadores generales)

	RAE			AE			SE		
	media	(	)	media	(	)	media	(	)
Population									
F	2.324	2.298	2.350	0.236	0.235	0.238	0.120	0.119	0.121
D	3.392	3.358	3.426	0.320	0.318	0.321	0.193	0.191	0.194

Tabla B.3: Por población

	RAE			AE			SE		
	media	(	)	media	(	)	media	(	)
Train n_samples									
5000	2.431	2.403	2.458	0.248	0.246	0.249	0.134	0.132	0.135
500	3.286	3.253	3.318	0.309	0.307	0.310	0.179	0.178	0.180

Tabla B.4: Por tamaño de muestra de entrenamiento

	RAE			AE			SE		
	media	(	)	media	(	)	media	(	)
Test n_samples									
10	1.731	1.718	1.744	0.301	0.300	0.303	0.175	0.174	0.176
100	3.986	3.945	4.026	0.255	0.253	0.256	0.138	0.137	0.139

Tabla B.5: Por tamaño de muestra de evaluación

	RAE			AE			SE		
	media	(	)	media	(	)	media	(	)
Train prev +									
0.50	1.865	1.833	1.896	0.180	0.179	0.182	0.073	0.072	0.074
0.25	2.105	2.067	2.142	0.220	0.218	0.222	0.106	0.104	0.107
0.75	2.162	2.124	2.199	0.217	0.216	0.219	0.103	0.102	0.105
0.01	4.037	3.977	4.098	0.382	0.380	0.385	0.246	0.244	0.248
0.99	4.123	4.061	4.184	0.390	0.387	0.392	0.254	0.252	0.256

Tabla B.6: Por prevalencia de clase positiva en muestra de entrenamiento

	RAE			AE			SE		
	media	(	)	media	(	)	media	(	)
Test prev +									
0.50	0.492	0.489	0.495	0.260	0.259	0.262	0.104	0.103	0.105
0.25	0.624	0.618	0.630	0.238	0.236	0.240	0.104	0.102	0.106
0.75	0.699	0.693	0.706	0.267	0.264	0.269	0.127	0.125	0.129
0.20	0.718	0.712	0.724	0.277	0.275	0.280	0.128	0.126	0.130
0.80	0.806	0.798	0.813	0.311	0.308	0.314	0.164	0.161	0.166
1.00	3.230	3.188	3.272	0.308	0.304	0.312	0.232	0.228	0.236
0.00	3.378	3.338	3.419	0.322	0.319	0.326	0.232	0.228	0.236
0.01	8.627	8.508	8.745	0.255	0.251	0.258	0.170	0.167	0.173
0.99	9.517	9.390	9.643	0.281	0.278	0.285	0.199	0.196	0.203

Tabla B.7: Por prevalencia de clase positiva en muestra de prueba

	RAE			AE			SE		
	media	(	)	media	(	)	media	(	)
Abs Dataset Shift									
(0-0.25]	1.100	1.085	1.114	0.191	0.190	0.192	0.078	0.077	0.079
(0.25-0.5]	1.774	1.747	1.800	0.259	0.258	0.261	0.117	0.116	0.118
[0]	1.846	1.790	1.902	0.139	0.137	0.141	0.058	0.056	0.059
(0.5-0.75]	3.878	3.822	3.933	0.361	0.359	0.364	0.220	0.218	0.222
(0.75-1.0]	9.956	9.844	10.067	0.604	0.601	0.607	0.484	0.480	0.487

Tabla B.8: Por *dataset shift* absoluto

		RAE			AE			SE		
		media	(	)	media	(	)	media	(	)
Abs Dataset Shift	**Cuantificador**									
[**0**]	**PCC**	0.070	0.067	0.072	0.020	0.020	0.021	0.001	0.001	0.001
	CC	0.269	0.259	0.280	0.074	0.069	0.078	0.013	0.011	0.014
	HDx	0.293	0.270	0.315	0.090	0.081	0.099	0.021	0.017	0.025
	ELM-SVMKLD	0.307	0.291	0.323	0.086	0.078	0.093	0.017	0.014	0.019
	ELM-SVMAE	0.348	0.332	0.364	0.101	0.093	0.109	0.020	0.018	0.023
	ELM-SVMRAE	0.374	0.356	0.391	0.111	0.102	0.120	0.024	0.022	0.027
	EMQ	0.786	0.700	0.872	0.109	0.105	0.113	0.036	0.033	0.039
	TH_MS	0.789	0.685	0.893	0.117	0.109	0.125	0.033	0.029	0.038
	TH_MAX	0.831	0.686	0.976	0.124	0.116	0.133	0.039	0.034	0.044
	TH_X	0.994	0.813	1.175	0.144	0.134	0.154	0.053	0.046	0.060
	TH_T50	1.674	1.369	1.980	0.171	0.159	0.183	0.074	0.064	0.083
	LR-Implicit	2.451	2.295	2.607	0.165	0.160	0.171	0.070	0.066	0.074
	PACC	3.169	3.011	3.326	0.192	0.186	0.197	0.077	0.074	0.080
	HDy	3.626	3.388	3.863	0.211	0.204	0.218	0.123	0.116	0.129
	ACC	5.205	4.763	5.646	0.273	0.260	0.285	0.122	0.113	0.131
(**0-0.25**]	**ELM-SVMRAE**	0.489	0.472	0.505	0.199	0.191	0.208	0.072	0.068	0.076
	CC	0.534	0.513	0.554	0.159	0.154	0.163	0.045	0.043	0.047
	ELM-SVMAE	0.554	0.528	0.579	0.192	0.184	0.200	0.065	0.061	0.068
	HDx	0.717	0.658	0.777	0.147	0.141	0.152	0.037	0.034	0.040
	EMQ	0.863	0.831	0.895	0.165	0.162	0.167	0.063	0.061	0.065
	ELM-SVMKLD	0.992	0.898	1.086	0.190	0.183	0.197	0.059	0.056	0.063
	TH_MAX	1.030	0.955	1.105	0.174	0.168	0.181	0.069	0.064	0.074
	TH_MS	1.051	0.973	1.129	0.172	0.165	0.178	0.067	0.062	0.072
	LR-Implicit	1.114	1.073	1.155	0.193	0.190	0.196	0.083	0.081	0.086
	PCC	1.135	1.108	1.162	0.155	0.154	0.156	0.031	0.031	0.032
	TH_X	1.157	1.052	1.262	0.185	0.178	0.192	0.078	0.073	0.084
	HDy	1.256	1.214	1.298	0.239	0.235	0.243	0.131	0.128	0.135
	PACC	1.256	1.212	1.299	0.202	0.199	0.205	0.083	0.081	0.086
	TH_T50	1.451	1.340	1.562	0.240	0.231	0.249	0.128	0.120	0.136
	ACC	1.578	1.465	1.691	0.247	0.240	0.254	0.111	0.105	0.117
(**0.25-0.5**]	**ACC**	1.235	1.083	1.388	0.149	0.140	0.157	0.071	0.064	0.078
	TH_MS	1.256	1.120	1.391	0.209	0.200	0.218	0.093	0.086	0.100
	PACC	1.283	1.219	1.347	0.177	0.173	0.181	0.078	0.075	0.081
	LR-Implicit	1.336	1.275	1.397	0.208	0.204	0.212	0.093	0.090	0.096
	TH_MAX	1.380	1.248	1.512	0.219	0.210	0.227	0.093	0.087	0.100
	EMQ	1.387	1.327	1.446	0.246	0.243	0.250	0.110	0.107	0.113
	HDx	1.461	1.302	1.620	0.254	0.244	0.265	0.099	0.094	0.105
	TH_T50	1.481	1.323	1.639	0.244	0.234	0.254	0.118	0.110	0.125
	HDy	1.598	1.530	1.665	0.275	0.271	0.280	0.132	0.129	0.136
	TH_X	1.623	1.448	1.797	0.240	0.230	0.249	0.114	0.106	0.122
	CC	2.622	2.479	2.764	0.361	0.354	0.369	0.165	0.159	0.170
	PCC	3.078	3.003	3.153	0.368	0.366	0.370	0.152	0.150	0.153
	ELM-SVMKLD	3.149	2.913	3.384	0.406	0.396	0.416	0.197	0.189	0.205
	ELM-SVMAE	3.216	2.980	3.453	0.426	0.415	0.437	0.217	0.209	0.226
	ELM-SVMRAE	3.229	2.994	3.464	0.434	0.422	0.445	0.228	0.218	0.238
(**0.5-0.75**]	**TH_T50**	2.067	1.839	2.294	0.248	0.234	0.262	0.141	0.129	0.153
	TH_X	2.182	1.949	2.416	0.254	0.240	0.267	0.141	0.129	0.152
	TH_MS	2.203	1.943	2.464	0.249	0.234	0.263	0.149	0.136	0.161
	PACC	2.213	2.113	2.314	0.240	0.234	0.245	0.122	0.117	0.126
	TH_MAX	2.263	2.048	2.479	0.255	0.242	0.268	0.136	0.125	0.147
	LR-Implicit	2.554	2.451	2.658	0.264	0.258	0.270	0.136	0.131	0.141
	EMQ	2.773	2.665	2.881	0.302	0.295	0.308	0.171	0.166	0.176
	ACC	3.137	2.831	3.443	0.284	0.270	0.298	0.160	0.147	0.174
	HDx	3.461	3.085	3.836	0.358	0.339	0.377	0.203	0.188	0.218
	HDy	3.607	3.474	3.739	0.357	0.350	0.363	0.221	0.215	0.228
	PCC	6.833	6.670	6.995	0.555	0.552	0.559	0.337	0.333	0.341
	CC	8.079	7.616	8.542	0.640	0.629	0.652	0.464	0.450	0.478
	ELM-SVMKLD	9.861	9.105	10.617	0.748	0.735	0.761	0.596	0.577	0.614
	ELM-SVMAE	11.143	10.315	11.971	0.831	0.823	0.840	0.705	0.692	0.719
	ELM-SVMRAE	11.791	10.912	12.670	0.870	0.862	0.878	0.771	0.757	0.784
(**0.75-1.0**]	**TH_T50**	6.358	5.779	6.937	0.383	0.363	0.404	0.283	0.261	0.304
	ACC	7.465	7.035	7.894	0.435	0.424	0.446	0.227	0.216	0.239
	PACC	7.539	7.316	7.762	0.455	0.448	0.463	0.287	0.280	0.295
	LR-Implicit	7.802	7.564	8.040	0.478	0.470	0.486	0.327	0.319	0.335
	TH_MS	8.124	7.500	8.748	0.487	0.466	0.509	0.377	0.355	0.399
	TH_MAX	8.515	7.935	9.095	0.507	0.488	0.525	0.368	0.348	0.388
	TH_X	8.591	7.976	9.206	0.524	0.503	0.544	0.404	0.383	0.425
	HDy	9.256	8.956	9.557	0.557	0.547	0.567	0.466	0.456	0.476
	EMQ	10.005	9.717	10.292	0.620	0.612	0.629	0.509	0.499	0.518
	HDx	12.655	11.704	13.606	0.750	0.731	0.769	0.618	0.594	0.642
	PCC	14.486	14.165	14.807	0.877	0.874	0.880	0.786	0.781	0.791
	CC	14.692	13.964	15.421	0.889	0.882	0.897	0.809	0.797	0.822
	ELM-SVMAE	15.348	14.282	16.414	0.930	0.923	0.937	0.873	0.860	0.887
	ELM-SVMKLD	15.348	14.282	16.414	0.930	0.923	0.937	0.873	0.860	0.887
	ELM-SVMRAE	15.348	14.282	16.414	0.930	0.923	0.937	0.873	0.860	0.887

Tabla B.9: Por *dataset shift* absoluto, y método de cuantificación

	RAE			AE			SE			F1			ECE		
	media	(	)	media	(	)	media	(	)	media	(	)	media	(	)
Calibración															
IsotonicCalibration	2.570	2.519	2.621	0.255	0.252	0.257	0.136	0.134	0.138	0.449	0.445	0.452	32.787	32.546	33.028
No Calibration	2.730	2.677	2.782	0.273	0.270	0.275	0.150	0.147	0.152	0.455	0.452	0.459	32.554	32.309	32.799
PlattScaling	2.850	2.795	2.905	0.279	0.276	0.281	0.156	0.154	0.158	0.451	0.447	0.454	31.715	31.471	31.959
BBQCalibration	2.951	2.895	3.007	0.281	0.279	0.284	0.159	0.157	0.162	0.449	0.446	0.453	33.197	32.943	33.451
HistogramCalibration	2.987	2.931	3.042	0.279	0.277	0.282	0.156	0.154	0.158	0.450	0.446	0.453	33.853	33.608	34.097

Tabla B.10: Por método de calibración (considerando solo métodos de cuantificación agregativos con clasificadores generales y blandos)

		RAE			AE			SE			F1			ECE		
		media	(	)	media	(	)	media	(	)	media	(	)	media	(	)
Cuantificador	**Calibración**															
LR-Implicit	**IsotonicCalibration**	2.096	1.998	2.195	0.219	0.214	0.224	0.109	0.105	0.113	0.451	0.443	0.458	32.709	32.168	33.251
TH_MS	**-**	2.102	1.995	2.208	0.224	0.219	0.229	0.120	0.115	0.124	0.451	0.444	0.459	NaN	NaN	NaN
EMQ	**IsotonicCalibration**	2.104	2.005	2.203	0.218	0.213	0.223	0.107	0.103	0.111	0.451	0.443	0.458	32.286	31.749	32.823
	No Calibration	2.130	2.031	2.228	0.239	0.234	0.244	0.122	0.117	0.126	0.458	0.450	0.465	32.377	31.830	32.924
TH_T50	**-**	2.172	2.065	2.279	0.251	0.246	0.257	0.140	0.135	0.144	0.451	0.444	0.459	NaN	NaN	NaN
PACC	**IsotonicCalibration**	2.183	2.080	2.286	0.233	0.228	0.238	0.119	0.114	0.123	0.446	0.439	0.454	33.224	32.686	33.762
TH_MAX	**-**	2.186	2.085	2.287	0.232	0.227	0.237	0.118	0.114	0.122	0.449	0.442	0.457	NaN	NaN	NaN
PACC	**PlattScaling**	2.188	2.088	2.289	0.229	0.224	0.234	0.116	0.111	0.120	0.446	0.439	0.454	31.799	31.255	32.343
LR-Implicit	**No Calibration**	2.193	2.092	2.294	0.244	0.239	0.249	0.126	0.122	0.130	0.458	0.450	0.465	32.377	31.830	32.924
PACC	**No Calibration**	2.227	2.123	2.331	0.229	0.224	0.234	0.116	0.112	0.120	0.453	0.446	0.460	32.754	32.207	33.302
TH_X	**-**	2.305	2.193	2.418	0.245	0.240	0.250	0.133	0.128	0.138	0.451	0.444	0.459	NaN	NaN	NaN
LR-Implicit	**PlattScaling**	2.424	2.310	2.537	0.253	0.248	0.258	0.139	0.134	0.144	0.454	0.446	0.461	31.690	31.146	32.235
	BBQCalibration	2.444	2.337	2.552	0.243	0.238	0.248	0.126	0.122	0.130	0.452	0.445	0.460	33.319	32.751	33.887
EMQ	**PlattScaling**	2.464	2.349	2.578	0.256	0.251	0.262	0.141	0.136	0.146	0.454	0.446	0.461	31.559	31.013	32.105
HDy	**IsotonicCalibration**	2.551	2.431	2.672	0.270	0.265	0.276	0.157	0.152	0.163	0.444	0.437	0.452	33.402	32.862	33.943
EMQ	**BBQCalibration**	2.574	2.446	2.703	0.281	0.275	0.287	0.173	0.168	0.179	0.450	0.443	0.458	33.284	32.717	33.852
	HistogramCalibration	2.639	2.515	2.762	0.277	0.271	0.282	0.166	0.160	0.171	0.451	0.443	0.458	33.591	33.044	34.137
LR-Implicit	**HistogramCalibration**	2.647	2.543	2.751	0.235	0.230	0.239	0.109	0.105	0.113	0.451	0.443	0.458	33.681	33.135	34.227
PACC	**HistogramCalibration**	2.653	2.547	2.759	0.238	0.233	0.242	0.113	0.109	0.116	0.448	0.441	0.456	34.147	33.602	34.693
HDx	**-**	2.716	2.544	2.888	0.272	0.264	0.279	0.146	0.140	0.153	NaN	NaN	NaN	NaN	NaN	NaN
PACC	**BBQCalibration**	2.744	2.643	2.845	0.229	0.224	0.233	0.097	0.094	0.100	0.447	0.439	0.454	33.329	32.761	33.897
HDy	**BBQCalibration**	2.881	2.743	3.018	0.303	0.297	0.309	0.196	0.189	0.202	0.446	0.439	0.454	33.089	32.520	33.657
ACC	**-**	2.887	2.775	2.999	0.255	0.250	0.260	0.125	0.121	0.129	0.452	0.445	0.460	NaN	NaN	NaN
HDy	**HistogramCalibration**	2.950	2.812	3.087	0.304	0.298	0.310	0.194	0.188	0.199	0.447	0.439	0.454	34.219	33.671	34.767
	No Calibration	3.142	3.008	3.275	0.311	0.305	0.317	0.187	0.182	0.193	0.451	0.443	0.458	32.884	32.335	33.433
	PlattScaling	3.269	3.134	3.403	0.319	0.313	0.325	0.191	0.186	0.197	0.448	0.440	0.455	31.964	31.418	32.509
PCC	**PlattScaling**	3.906	3.765	4.048	0.337	0.331	0.342	0.193	0.188	0.198	0.452	0.444	0.459	31.561	31.013	32.109
CC	**-**	3.909	3.757	4.061	0.362	0.356	0.368	0.228	0.222	0.234	0.458	0.450	0.465	NaN	NaN	NaN
PCC	**IsotonicCalibration**	3.914	3.775	4.054	0.333	0.328	0.339	0.188	0.183	0.193	0.451	0.444	0.459	32.314	31.780	32.848
	No Calibration	3.957	3.814	4.100	0.340	0.335	0.346	0.197	0.191	0.202	0.458	0.450	0.465	32.377	31.830	32.924
	HistogramCalibration	4.046	3.902	4.190	0.344	0.338	0.349	0.199	0.193	0.204	0.452	0.444	0.459	33.626	33.080	34.171
	BBQCalibration	4.112	3.967	4.258	0.349	0.344	0.355	0.204	0.199	0.210	0.451	0.443	0.458	32.963	32.399	33.528
ELM-SVMKLD	**-**	4.569	4.335	4.803	0.407	0.398	0.416	0.271	0.262	0.280	0.385	0.375	0.396	NaN	NaN	NaN
ELM-SVMAE	**-**	4.638	4.394	4.882	0.428	0.419	0.437	0.296	0.286	0.305	0.374	0.363	0.385	NaN	NaN	NaN
ELM-SVMRAE	**-**	4.724	4.473	4.975	0.440	0.430	0.449	0.312	0.302	0.321	0.357	0.346	0.368	NaN	NaN	NaN

Tabla B.11: Por método de cuantificación y de calibración (considerando solo métodos de cuantificación agregativos con clasificadores generales y blandos)

Train prev +	Calibración	RAE media	(	)	AE media	(	)	SE media	(	)	F1 media	(	)	ECE media	(	)
0.01	**IsotonicCalibration**	3.575	3.430	3.720	0.352	0.346	0.358	0.217	0.211	0.222	0.075	0.071	0.078	45.800	45.140	46.460
	No Calibration	3.902	3.748	4.055	0.391	0.385	0.397	0.255	0.249	0.261	0.083	0.079	0.087	46.424	45.754	47.094
	PlattScaling	4.018	3.860	4.175	0.387	0.381	0.394	0.252	0.246	0.258	0.079	0.075	0.083	45.715	45.055	46.375
	BBQCalibration	4.376	4.217	4.536	0.391	0.385	0.397	0.250	0.244	0.256	0.078	0.074	0.082	47.172	46.493	47.851
	HistogramCalibration	4.398	4.238	4.557	0.392	0.386	0.398	0.251	0.245	0.257	0.075	0.071	0.079	46.650	45.977	47.323
0.25	**IsotonicCalibration**	1.828	1.747	1.909	0.192	0.188	0.196	0.082	0.078	0.085	0.351	0.345	0.358	25.940	25.591	26.289
	BBQCalibration	1.909	1.821	1.997	0.206	0.202	0.211	0.096	0.092	0.099	0.350	0.344	0.357	26.919	26.541	27.298
	HistogramCalibration	2.010	1.922	2.098	0.205	0.201	0.209	0.092	0.088	0.095	0.354	0.347	0.360	27.190	26.831	27.549
	No Calibration	2.069	1.980	2.159	0.213	0.209	0.218	0.095	0.092	0.099	0.371	0.364	0.377	26.285	25.932	26.638
	PlattScaling	2.113	2.022	2.204	0.219	0.214	0.223	0.100	0.096	0.103	0.353	0.347	0.360	25.085	24.732	25.439
0.50	**No Calibration**	1.789	1.712	1.865	0.171	0.167	0.175	0.065	0.062	0.068	0.579	0.573	0.586	15.390	15.217	15.564
	IsotonicCalibration	1.791	1.711	1.871	0.179	0.175	0.183	0.073	0.070	0.077	0.576	0.569	0.582	18.639	18.423	18.856
	BBQCalibration	1.821	1.739	1.903	0.181	0.177	0.185	0.075	0.072	0.079	0.577	0.571	0.583	15.042	14.830	15.253
	PlattScaling	1.869	1.788	1.950	0.178	0.174	0.182	0.072	0.069	0.075	0.578	0.572	0.585	14.550	14.372	14.727
	HistogramCalibration	1.874	1.793	1.954	0.183	0.179	0.187	0.075	0.072	0.078	0.577	0.571	0.584	19.380	19.156	19.604
0.75	**IsotonicCalibration**	1.904	1.820	1.988	0.189	0.185	0.193	0.081	0.077	0.084	0.614	0.607	0.621	26.072	25.711	26.434
	HistogramCalibration	2.056	1.969	2.143	0.203	0.198	0.207	0.090	0.086	0.093	0.614	0.607	0.621	27.884	27.508	28.260
	BBQCalibration	2.080	1.987	2.172	0.209	0.204	0.213	0.096	0.092	0.100	0.614	0.607	0.621	28.032	27.631	28.434
	No Calibration	2.081	1.993	2.168	0.207	0.203	0.212	0.088	0.085	0.092	0.616	0.609	0.623	26.808	26.448	27.169
	PlattScaling	2.150	2.059	2.242	0.215	0.210	0.219	0.096	0.093	0.100	0.615	0.608	0.622	25.712	25.351	26.073
0.99	**IsotonicCalibration**	3.752	3.601	3.902	0.361	0.354	0.367	0.228	0.222	0.234	0.585	0.578	0.592	47.484	46.810	48.158
	No Calibration	3.808	3.656	3.960	0.381	0.375	0.387	0.244	0.238	0.250	0.586	0.579	0.593	47.863	47.185	48.541
	PlattScaling	4.101	3.942	4.260	0.396	0.390	0.402	0.260	0.254	0.266	0.585	0.578	0.592	47.512	46.840	48.184
	BBQCalibration	4.570	4.403	4.736	0.418	0.412	0.424	0.279	0.273	0.286	0.585	0.578	0.592	48.819	48.127	49.511
	HistogramCalibration	4.597	4.432	4.762	0.414	0.408	0.420	0.273	0.266	0.279	0.585	0.578	0.592	48.159	47.476	48.843

Tabla B.12: Por prevalencia de clase positiva en muestra de entrenamiento y de calibración (considerando solo métodos de cuantificación agregativos con clasificadores generales y blandos)

Referencias

[1] George Forman. Counting positives accurately despite inaccurate classification. In *European conference on machine learning*, pages 564–575. Springer, 2005.

[2] David D Lewis. Evaluating and optimizing autonomous text classification systems. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 246–254, 1995.

[3] Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern recognition*, 45(1):521–530, 2012.

[4] Amos Storkey et al. When training and test sets are different: characterizing learning transfer. *Dataset shift in machine learning*, 30(3-28):6, 2009.

[5] Rocío Alaíz-Rodríguez, Alicia Guerrero-Curieses, and Jesús Cid-Sueiro. Class and subclass probability re-estimation to adapt a classifier in the presence of concept drift. *Neurocomputing*, 74(16):2614–2623, 2011.

[6] Marthinus Christoffel Du Plessis and Masashi Sugiyama. Class prior estimation from positive and unlabeled data. *IEICE TRANSACTIONS on Information and Systems*, 97(5):1358–1362, 2014.

[7] Yee Seng Chan and Hwee Tou Ng. Estimating class priors in domain adaptation for word sense disambiguation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 89–96, 2006.

[8] Zhihao Zhang and Jie Zhou. Transfer estimation of evolving class priors in data stream classification. *Pattern Recognition*, 43(9):3151–3161, 2010.

[9] Marthinus Christoffel Du Plessis and Masashi Sugiyama. Semi-supervised learning of class balance under class-prior change by distribution matching. *Neural Networks*, 50:110–119, 2014.

[10] Jose Barranquero, Pablo González, Jorge Díez, and Juan José Del Coz. On the study of nearest neighbor algorithms for prevalence estimation in binary problems. *Pattern Recognition*, 46(2): 472–482, 2013.

[11] Hideki Asoh, Kazushi Ikeda, and Chihiro Ono. A fast and simple method for profiling a population of twitter users. In *The Third International Workshop on Mining Ubiquitous and Social Environments*, page 19. Citeseer, 2012.

[12] Víctor González-Castro, Rocío Alaiz-Rodríguez, and Enrique Alegre. Class distribution estimation based on the Hellinger distance. *Information Sciences*, 218:146–164, 2013.

[13] Nachai Limsetto and Kitsana Waiyamai. Handling concept drift via ensemble and class distribution estimation technique. In *Advanced Data Mining and Applications: 7th International Conference, ADMA 2011, Beijing, China, December 17-19, 2011, Proceedings, Part II 7*, pages 13–26. Springer, 2011.

[14] Jack Chongjie Xue and Gary M Weiss. Quantification and semi-supervised classification methods for handling changes in class distribution. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 897–906, 2009.

[15] Afonso Fernandes Vaz, Rafael Izbicki, and Rafael Bassi Stern. Quantification under prior probability shift: The ratio estimator and its extensions. *The Journal of Machine Learning Research*, 20(1):2921–2953, 2019.

[16] Andrea Esuli, Alessandro Fabris, Alejandro Moreo, and Fabrizio Sebastiani. *Learning to Quantify*. Springer Nature, 2023.

[17] Pablo González, Alberto Castaño, Nitesh V Chawla, and Juan José Del Coz. A review on quantification learning. *ACM Computing Surveys (CSUR)*, 50(5):1–40, 2017.

[18] George Forman. Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17:164–206, 2008.

[19] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[20] Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1):21–41, 2002.

[21] Arpita Biswas and Suvam Mukherjee. Ensuring fairness under prior probability shifts. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 414–424, 2021.

[22] Alessandro Fabris, Andrea Esuli, Alejandro Moreo, and Fabrizio Sebastiani. Measuring fairness under unawareness of sensitive attributes: A quantification-based approach. *Journal of Artificial Intelligence Research*, 76:1117–1180, 2023.

[23] Isabelle M Guyon. Design of experiments for the NIPS 2003 variable selection benchmark. In *Feature Extraction. Studies in Fuzziness and Soft Computing*, 2003. URL `https://api.semanticscholar.org/CorpusID:115452637`.

[24] Antonio Bella, Cesar Ferri, José Hernández-Orallo, and Maria Jose Ramirez-Quintana. Quantification via probability estimators. In *2010 IEEE International Conference on Data Mining*, pages 737–742. IEEE, 2010.

[25] Lei Tang, Huiji Gao, and Huan Liu. Network quantification despite biased labels. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, pages 147–154, 2010.

[26] Dirk Tasche. Exact fit of simple finite mixture models. *Journal of Risk and Financial Management*, 7(4):150–164, 2014.

[27] George Forman. Quantifying trends accurately despite classifier error and class imbalance. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 157–166, 2006.

[28] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.

[29] Andrea Esuli, Alessio Molinari, and Fabrizio Sebastiani. A critical reassessment of the Saerens-Latinne-Decaestecker algorithm for posterior probability adjustment. *ACM Transactions on Information Systems (TOIS)*, 39(2):1–34, 2020.

[30] Amr Alexandari, Anshul Kundaje, and Avanti Shrikumar. Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In *International Conference on Machine Learning*, pages 222–232. PMLR, 2020.

[31] Andrea Esuli, Fabrizio Sebastiani, and Ahmed Abbasi. Sentiment Quantification. *IEEE Intell. Syst.*, 25(4):72–75, 2010.

[32] Andrea Esuli and Fabrizio Sebastiani. Explicit Loss Minimization in Quantification Applications (Preliminary Draft). In *DART@ AI* IA*, pages 1–11. Citeseer, 2014.

[33] Andrea Esuli and Fabrizio Sebastiani. Optimizing text quantifiers for multivariate loss functions. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(4):1–27, 2015.

[34] Thorsten Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd international conference on Machine learning*, pages 377–384, 2005.

[35] Jose Barranquero, Jorge Díez, and Juan José del Coz. Quantification-oriented learning based on reliable classifiers. *Pattern Recognition*, 48(2):591–604, 2015.

[36] Alejandro Moreo and Fabrizio Sebastiani. Re-assessing the "classify and count" quantification method. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II 43*, pages 75–91. Springer, 2021.

[37] Purushottam Kar, Shuai Li, Harikrishna Narasimhan, Sanjay Chawla, and Fabrizio Sebastiani. Online optimization methods for the quantification problem. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1625–1634, 2016.

[38] Amartya Sanyal, Pawan Kumar, Purushottam Kar, Sanjay Chawla, and Fabrizio Sebastiani. Optimizing non-decomposable measures with deep networks. *Machine Learning*, 107:1597–1620, 2018.

[39] Katherine Keith and Brendan O'Connor. Uncertainty-aware generative models for inferring document class prevalence. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4575–4585, 2018.

[40] Fabrizio Sebastiani. Evaluation measures for quantification: An axiomatic approach. *Information Retrieval Journal*, 23(3):255–288, 2020.

[41] Alejandro Moreo and Fabrizio Sebastiani. Tweet sentiment quantification: An experimental re-evaluation. *PLoS One*, 17(9):e0263449, 2022.

[42] Waqar Hassan, André Gustavo Maletzke, and Gustavo Enrique de Almeida Prado Alves Batista. Pitfalls in quantification assessment. In *Proceedings*, 2021.

[43] Alejandro Moreo, Andrea Esuli, and Fabrizio Sebastiani. QuaPy: A Python-based framework for quantification. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 4534–4543, 2021.

[44] Dirk Tasche. Does quantification without adjustments work? *arXiv preprint arXiv:1602.08780*, 2016.

[45] Tobias Schumacher, Markus Strohmaier, and Florian Lemmerich. A comparative evaluation of quantification methods. *arXiv preprint arXiv:2103.03223*, 2021.

[46] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[47] Fabian Kuppers, Jan Kronenberger, Amirhossein Shantia, and Anselm Haselhoff. Multivariate confidence calibration for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 326–327, 2020.

[48] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[49] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699, 2002.

[50] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632, 2005.

[51] Geoffrey Stewart Morrison. Tutorial on logistic-regression calibration and fusion: converting a score to a likelihood ratio. *Australian Journal of Forensic Sciences*, 45(2):173–197, 2013.