Análisis de algoritmos de cuantificación basados en ajuste de distribuciones*

Alberto Castaño
Centro de Inteligencia Artificial
Universidad de Oviedo
Gijón, España
castanoalberto@uniovi.es

Laura Morán-Fernández
Departamento de Computación
Universidade da Coruña
A Coruña, España
laura.moranf@udc.es

Jaime Alonso
Centro de Inteligencia Artificial
Universidad de Oviedo
Gijón, España
jalonso@uniovi.es

Verónica Bolón-Canedo
Departamento de Computación
Universidade da Coruña
A Coruña, España
veronica.bolon@udc.es

Amparo Alonso-Betanzos
Departamento de Computación
Universidade da Coruña
A Coruña, España
ciamparo@udc.es

Juan José del Coz
Centro de Inteligencia Artificial
Universidad de Oviedo
Gijón, España
juanjo@uniovi.es

***Resumen*—La cuantificación consiste en predecir la proporción de las distintas clases presentes en una muestra dada de ejemplos. Un tipo de algoritmos de cuantificación se basa en estimar y ajustar las distribuciones subyacentes del conjunto de entrenamiento y de la muestra a predecir. La diferencia clave entre los métodos propuestos reside en cómo estimar las distribuciones: usando los propios atributos o las predicciones dadas por un clasificador. Este artículo analiza ambas alternativas y propone dos nuevos algoritmos. La conclusión es que los métodos basados en usar las predicciones de un clasificador no son en general peores y en ciertos casos producen mejores estimaciones.**

***Palabras clave:*—Cuantificación, Estimación de la prevalencia, Adaptación al dominio**

I. Introducción

Existen diversas aplicaciones reales que demandan predecir la distribución de probabilidad de las clases en un conjunto de ejemplos. Formalmente, este problema se denomina cuantificación [1]. Ejemplos típicos son predecir la proporción de comentarios positivos y negativos en una red social sobre un acontecimiento o producto durante un período de tiempo [2], cuantificar la cantidad de células dañadas en un tejido [3] o predecir el porcentaje de incidencias de cada tipo en un centro de atención al usuario [1]. En estas aplicaciones no se requiere dar una predicción individual para cada ejemplo sino que basta con retornar una predicción única para toda la muestra.

Aunque a primera vista la cuantificación pueda verse como un subproducto de la clasificación, se pueden diseñar algoritmos de cuantificación que no se basen simplemente en agregar las predicciones dadas por un clasificador. Es el caso de los métodos que estiman y ajustan distribuciones de muestras de ejemplos. Si nos centramos en la cuantificación binaria, la idea consiste en estimar durante el entrenamiento la distribución de la clase positiva y negativa de alguna forma y, en el momento de predecir una nueva muestra, estimar de la misma manera su distribución y aproximarla con una combinación de la distribución de la clase positiva y negativa dependiendo de la proporción de cada clase. La proporción que más aproxime ambas distribuciones será la que retorne el método [3], [4].

Se han propuesto sistemas basados en el ajuste de distribuciones no sólo en el campo de la cuantificación [3], sino también en otro problema relacionado como es el de los algoritmos de adaptación al dominio [5]. En estos últimos la idea es calcular la probabilidad a priori de las clases para actualizar un clasificador, sin necesidad de reentrenarlo, cuando la distribución de las clases cambia. El primer objetivo de este artículo es aunar y analizar ambas líneas de trabajo. Una diferencia clave entre ambas corrientes es cómo estimar las distribuciones. Mientras los algoritmos de adaptación al dominio se basan en estimar las distribuciones usando la información con la que se describen los ejemplos, los métodos de cuantificación proponen usar también las predicciones de un clasificador, asumiendo que ejemplos similares deben obtener predicciones parecidas. Un segundo objetivo de este artículo es comparar ambas alterativas. Asimismo, proponemos dos nuevos métodos que extienden algoritmos de adaptación al dominio, pero usando las predicciones de un clasificador.

Con todo ello, las contribuciones de este trabajo son: 1) presentar dos nuevos cuantificadores basados en ajustar distribuciones usando las predicciones de un clasificador, y 2) realizar un análisis experimental riguroso de los métodos basados en ajuste de distribuciones.

II. Cuantificación Binaria

Sea $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ un conjunto de entrenamiento, donde $\boldsymbol{x}_i$ es la representación de un ejemplo en el espacio de entrada $\mathcal{X} \subset \mathbb{R}^d$, e $y_i \in \mathcal{Y} = \{-1, +1\}$ su clase. El objetivo de la cuantificación binaria es obtener un modelo, $\bar{h}$, que dado un conjunto de ejemplos sin etiquetar, $T = \{\boldsymbol{x}_j\}_{j=1}^{m}$,

*Esta investigación ha sido financiada en parte por el Ministerio de Economía y Competitividad (MINECO) y el Fondo Europeo de Desarrollo Regional (FEDER), a través del proyecto coordinado TIN2015-65069-C2.

prediga la proporción de la clase positiva y la clase negativa. Dado que ambas son complementarias, basta con predecir la proporción o prevalencia de la clase positiva $\hat{p}$, siendo la de la clase negativa $1-\hat{p}$. En símbolos: $\bar{h} : \mathbb{N}^{\mathcal{X}} \longrightarrow [0,1]$, donde $\mathbb{N}^{\mathcal{X}}$ denota un multi-conjunto de ejemplos, representado por el número de veces que cada posible ejemplo $\boldsymbol{x} \in \mathcal{X}$ aparece en dicho multi-conjunto.

La aproximación más sencilla para producir un cuantificador es la denominada Clasificar y Contar (CC), que consiste en entrenar un clasificador binario, h, clasificar con él todos los ejemplos de T y contar los ejemplos de cada clase: $\hat{p}_{CC} = \bar{h}(T) = \frac{1}{m}\sum_{\boldsymbol{x}_i \in D} I(h(\boldsymbol{x}_i) = +1)$, siendo I la función indicatriz. Sin embargo, los estudios muestran [6] que CC produce en general resultados subóptimos, especialmente cuando la distribución de las clases cambia significativamente entre el entrenamiento y la fase de predicción. Por ese motivo se han propuesto distintos algoritmos de cuantificación. Una revisión completa de ellos puede encontrarse en [7].

La cuantificación, por su propia definición, es uno de esos problemas de aprendizaje [8] en los que se sabe de antemano que la distribución de los datos cambia, es decir, que $P_D(x,y) \neq P_T(x,y)$, ya que es obvio que al menos cambia $P_D(y) \neq P_T(y)$[1]. La cuestión es determinar qué elementos de la distribución, entre $P(x)$, $P(x|y)$ y $P(y|x)$, se asume que no cambian para facilitar el aprendizaje. De hecho, la clave y el primer paso para diseñar un algoritmo de cuantificación es definir precisamente eso. La mayoría de los algoritmos de cuantificación están diseñados bajo la asunción de que $P(x|y)$ se mantiene constante y que el resto de factores puede cambiar.

Esta asunción la sigue por ejemplo el método AC [1], que probablemente es el cuantificador más usado y que tomaremos como algoritmo básico de comparación en este estudio. La idea del método AC es que, dado que $P(x|y)$ es constante, los ratios de verdaderos positivos, tpr, (*true positive rate*) y de falsos positivos, fpr, (*false positive rate*) del clasificador usado por el método CC serán constantes también, aunque cambie la distribución de probabilidad de las clases. Eso implica que la prevalencia que predice el método CC, $\hat{p}_{CC}$, se puede escribir en función de la prevalencia real p:

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

Despejando p calculamos la prevalencia estimada por AC:

$$\hat{p}_{AC} = \frac{\hat{p}_{CC} - fpr}{tpr - fpr}. \tag{2}$$

En base a este desarrollo matemático, los pasos del método AC consisten en, primero, entrenar un clasificador y estimar su *tpr* y *fpr*, después usar el método CC para calcular $\hat{p}_{CC}$ y finalmente aplicar (2) para obtener la predicción final, $\hat{p}_{AC}$. En teoría AC produce predicciones perfectas siempre que se cumpla que: 1) $P(x|y)$ es constante y 2) las estimaciones del *tpr* y *fpr* son perfectas. Obviamente, es difícil que ambas condiciones se cumplan totalmente en problemas reales de una cierta complejidad, pero aún así el método suele ofrecer un buen rendimiento.

[1]Nótese que si $P_D(y) = P_T(y)$ el problema de cuantificación sería trivial

III. Métodos Basados en Ajuste de Distribuciones

El enfoque para el problema de la cuantificación en el que se centra este artículo se basa en estimar y ajustar las distribuciones de entrenamiento y test. La idea fundamental consiste en modificar la distribución de entrenamiento, que abusando de notación denotaremos por D', mediante una mezcla de la distribución de los ejemplos positivos, D^+, y la distribución de los negativos, D^-, en función de la prevalencia estimada de ambas clases, es decir,

$$D' = D^- \cdot (1-\hat{p}) + D^+ \cdot \hat{p}. \tag{3}$$

El objetivo es tratar de aproximar D' lo más posible a la distribución estimada para el conjunto de test T. La Figura 1 trata de ilustrar esta idea. En la figura izquierda se observa la distribución de los ejemplos positivos y de los negativos de entrenamiento. En la parte derecha se muestra la distribución observable en el conjunto de test. Asumiendo de nuevo que $P(x|y)$ es constante, y que por tanto las distribuciones de D^+ y D^- cambiarán uniformemente en función de la prevalencia de las clases, el objetivo es minimizar la diferencia entre la distribución de test y la distribución modificada D':

$$\min_{\hat{p}} \Delta(T, D') = \min_{\hat{p}} \Delta(T, D^- \cdot (1-\hat{p}) + D^+ \cdot \hat{p}). \tag{4}$$

En el ejemplo de la Figura 1 es evidente que la prevalencia de la clase positiva en el conjunto de test es menor que la observada en los datos de entrenamiento y que por lo tanto, tenemos que disminuir ese valor para ajustar las distribuciones.

Los métodos basados en ajustar distribuciones comparten un marco común que consta de los siguientes elementos:

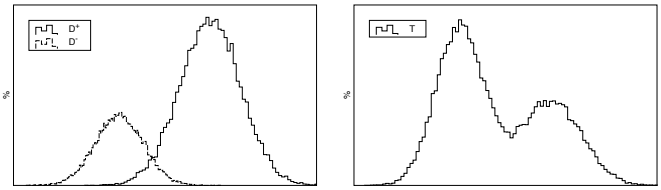
- una forma de estimar o representar las distribuciones,
- una medida Δ para compararlas, y
- un método para calcular el mínimo de (4).

La diferencia entre los métodos que se estudian en este artículo está en alguno de estos tres elementos, pero nuestro interés principal es analizar las distintas formas usadas para estimar las distribuciones y en menor medida las métricas empleadas para compararlas.

En la Figura 1 se ha dejado deliberadamente sin etiquetar el eje horizontal para indicar que la distribución se puede representar usando diferentes datos. No obstante, hay dos corrientes fundamentales: usar la propia descripción de los ejemplos (lo que algunos métodos llaman usar la información de las X's), o usar las predicciones dadas por un clasificador (usar las predicciones y)[2]. Las distribuciones de la Figura 1 podrían perfectamente representar funciones de densidad de probabilidad tanto del valor de un atributo del espacio de entrada $\mathcal{X}$, como de la probabilidad a posteriori dada por un clasificador de que un ejemplo pertenezca a la clase positiva, $P(y=+1|x)$.

En cuanto a la medida Δ, hay múltiples opciones incluyendo normas como $L1$ o $L2$ y medidas de similitud o divergencia entre distribuciones de probabilidad como la divergencia

[2]No confundir con usar las etiquetas reales presentes en D



(a) Distribuciones del cjto de entrenamiento (b) Distribución del cjto de test

Figura 1: Los métodos basados en comparación y ajuste estiman de alguna forma la distribución de (a) los ejemplos positivos y negativos disponibles en el entrenamiento, D^+ y D^-, y (b) de los ejemplos de la muestra a predecir, T. Después tratan de ajustar la combinación de las distribuciones de D^+ y D^- usando (3) para aproximar la distribución de T

de Kullback-Leibler (KLD) y la distancia de Hellinger. En el artículo analizaremos las medidas que se han propuesto hasta ahora y propondremos dos nuevos métodos usando una función basada en distancias y otra en rankings.

III-A. Ajuste Mediante la Distancia de Hellinger

Analizando la literatura de cuantificación, los métodos más relevantes basados en ajuste de distribuciones son los propuestos en [3]. Se trata de dos métodos basados en la distancia de Hellinger (HD) para comparar las distribuciones, donde éstas se representan mediante histogramas: en un caso del valor de los atributos (versión llamada HDX) y en el otro del valor de las predicciones de un clasificador (HDy). Usar histogramas hace que estos métodos tengan un hiperparámetro, b, que es el número de intervalos o *bins* usados en los histogramas.

Usando la definición de la distancia de Hellinger para el caso discreto multivariante, y aplicando (3) para representar D', resulta en el siguiente problema a minimizar:

$$\min_{\hat{p}} \frac{1}{d}\sum_{l=1}^{d}\sqrt{\sum_{k=1}^{b}\left(\sqrt{\frac{|T_{k,l}|}{m}}-\sqrt{\frac{|D^-_{k,l}|}{n^-}(1-\hat{p})+\frac{|D^+_{k,l}|}{n^+}\hat{p}}\right)^2}, \tag{5}$$

donde n^- y n^+ son el número de ejemplos en D^- y D^+ respectivamente y $|T_{k,l}|/m$, $|D^-_{k,l}|/n^-$ y $|D^+_{k,l}|/n^+$ son la proporción de ejemplos de T, D^- y D^+ que pertenece al bin k en la dimensión l. En el caso de HDy, solamente tenemos una dimensión que se corresponde con las predicciones del clasificador, luego el primer sumatorio desaparece.

Los autores proponen una búsqueda lineal, variando $\hat{p}$ en el rango $[0, 1]$ en pequeños incrementos, para resolver (5). Sin embargo, la solución se puede hallar analíticamente, con más precisión y menos coste computacional, teniendo en cuenta la equivalencia entre la distancia de Hellinger y el coeficiente de Bhattacharyya, $HD(T, D') = \sqrt{1 - BC(T, D')}$ [9], y resolviendo el siguiente problema de optimización:

$$\begin{aligned} \min_{\pi} \quad & 1-\sum_k \sqrt{T_k(D'\pi)_k}, \\ s.a. \quad & \sum \pi = 1, \quad \pi_i \geq 0, \end{aligned} \tag{6}$$

donde, sobrecargando la notación, definimos:

- T como una matriz con $bd \times 1$ en el caso del método HDX, con las proporciones de los bins del primer atributo en las b primeras filas, y así sucesivamente con el resto de atributos, y en el caso de HDy con solamente b filas y una columna con las proporciones de cada bin calculadas con las predicciones del clasificador,
- D' es una matriz de $bd \times 2$ en el caso del método HDX, y $b \times 2$ en el caso del HDy, con idéntica estructura a la matriz T, donde las dos columnas representan, en este orden, las proporciones de la clase negativa y positiva, y
- π una matriz 2×1 con dos variables, π_1 la prevalencia de la clase negativa, y π_2 la de la positiva, $\hat{p}$.

Este problema puede resolverse con cualquier librería de optimización convexa; nosotros empleamos `cvxpy` para Python.

En teoría, la ventaja de HDX es usar directamente la información disponible en los atributos con los que se representan los ejemplos, lo cual puede ser una desventaja en espacios de alta dimensionalidad, o cuando los problemas dependan mucho de la interacción entre los valores de los atributos, ya que estos se consideran de forma en cierto modo independiente al calcular la HD, como hemos descrito anteriormente. La ventaja del método HDy es que resume en una sola dimensión, a través de un clasificador, la distribución de los ejemplos, asumiendo que ejemplos parecidos deben obtener predicciones similares. La desventaja del método HDy es que el clasificador debe entrenarse adecuadamente y las estimaciones de las predicciones de los ejemplos deben hacerse no en reescritura, sino usando validaciones cruzadas con un alto número de particiones para evitar el sobreajuste en las predicciones.

III-B. Ajuste Mediante la Distancia Energy

Varios métodos de adaptación al dominio [5], [10] se basan en calcular la nueva distribución de las clases (denominadas probabilidades a priori en este contexto). La idea es usar esas nuevas probabilidades a priori para ajustar el clasificador disponible sin necesidad de reentrenarlo. Es obvio que dichos métodos también pueden usarse como algoritmos de cuantificación, aunque su objetivo sea otro bien distinto. En la literatura de adaptación al dominio es habitual que la comparación entre las distribuciones de entrenamiento y test se haga solamente usando el valor de los atributos [4], [11]. Dichos métodos plantean los mismos problemas que comentamos anteriormente para el método HDX: a veces es complejo calcular las densidades antes de compararlas. No obstante, siguiendo ese enfoque, recientemente se ha publicado un método [12] que destaca entre los propuestos en ese campo ya que es computacionalmente muy eficiente y tiene un mejor rendimiento que otros métodos anteriores [4].

Dicho algoritmo, que denotaremos por EDX, se basa en minimizar la distancia *Energy* (ED) entre la distribución de T y la distribución modificada D' definida por:

$$\begin{aligned} \min_{\hat{p}} \quad & 2\cdot\mathbb{E}_{\boldsymbol{x}_j\backsim T,\boldsymbol{x}_i\backsim D'}||\boldsymbol{x}_j-\boldsymbol{x}_i|| \\ & -\mathbb{E}_{\boldsymbol{x}_j,\boldsymbol{x}'_j\backsim T}||\boldsymbol{x}_j-\boldsymbol{x}'_j|| - \mathbb{E}_{\boldsymbol{x}_i,\boldsymbol{x}'_i\backsim D'}||\boldsymbol{x}_i-\boldsymbol{x}'_i||, \end{aligned} \tag{7}$$

donde $||\cdot||$ representa la distancia euclídea. Nótese que el segundo término puede suprimirse al ser independiente de $\hat{p}$ debido a que solo interviene la distribución de T. Omitimos aquí la derivación matemática para resolver este problema ya que es similar al que detallaremos en la sección siguiente.

IV. Métodos Propuestos

Nuestra primera propuesta consiste en hacer el método complementario al EDX, que llamaremos EDy y que emplea las predicciones de un clasificador h en lugar de los atributos que definen $\mathcal{X}$. Para ello debemos minimizar la expresión:

$$\begin{aligned}\min_{\hat{p}} \quad & 2 \cdot \mathbb{E}_{\boldsymbol{x}_j \backsim T, \boldsymbol{x}_i \backsim D'} ||h(\boldsymbol{x}_j) - h(\boldsymbol{x}_j)||_1 \\ & -\mathbb{E}_{\boldsymbol{x}_i, \boldsymbol{x}'_i \backsim D'} ||h(\boldsymbol{x}_i) - h(\boldsymbol{x}'_i)||_1.\end{aligned} \tag{8}$$

Si desdoblamos ambos términos aplicando (3) tenemos

$$\begin{aligned}\min_{\hat{p}} \quad & 2 \cdot (1-\hat{p}) \cdot \mathbb{E}_{\boldsymbol{x}_j \backsim T, \boldsymbol{x}_i \backsim D^-} ||h(\boldsymbol{x}_j) - h(\boldsymbol{x}_i)||_1 \\ & +2 \cdot \hat{p} \cdot \mathbb{E}_{\boldsymbol{x}_j \backsim T, \boldsymbol{x}_i \backsim D^+} ||h(\boldsymbol{x}_j) - h(\boldsymbol{x}_i)||_1 \\ & -(1-\hat{p})^2 \cdot \mathbb{E}_{\boldsymbol{x}_i, \boldsymbol{x}'_i \backsim D^-} ||h(\boldsymbol{x}_i) - h(\boldsymbol{x}'_i)||_1 \\ & -2 \cdot \hat{p} \cdot (1-\hat{p}) \mathbb{E}_{\boldsymbol{x}_i \backsim D^+, \boldsymbol{x}'_i \backsim D^-} ||h(\boldsymbol{x}_i) - h(\boldsymbol{x}'_i)||_1 \\ & -\hat{p}^2 \cdot \mathbb{E}_{\boldsymbol{x}_i, \boldsymbol{x}'_i \backsim D^+} ||h(\boldsymbol{x}_i) - h(\boldsymbol{x}'_i)||_1.\end{aligned} \tag{9}$$

En la práctica podemos aproximar cada esperanza usando las correspondientes muestras de ejemplos disponibles:

$$\mathbb{E}_{\substack{\boldsymbol{x}_j \backsim T \\ \boldsymbol{x}_i \backsim D^-}} \ldots \approx \mu_{T,D^-} = \frac{1}{mn^-} \sum_{\boldsymbol{x}_j \in T} \sum_{\boldsymbol{x}_i \in D^-} ||h(\boldsymbol{x}_j) - h(\boldsymbol{x}_i)||_1. \tag{10}$$

Sustituyendo todos los términos tenemos

$$\begin{aligned}\min_{\hat{p}} \quad & 2(1-\hat{p})\mu_{T,D^-} + 2\hat{p}\mu_{T,D^+} \\ & -(1-\hat{p})^2 \mu_{D^-,D^-} - 2\hat{p}(1-\hat{p})\mu_{D^-,D^+} - \hat{p}^2 \mu_{D^+,D^+},\end{aligned} \tag{11}$$

un problema fuertemente convexo [12] en el que derivando e igualando a cero obtenemos que

$$\hat{p} = \frac{\mu_{T,D^-} - \mu_{T,D^+} - \mu_{D^-,D^-} + \mu_{D^-,D^+}}{-\mu_{D^-,D^-} + 2\mu_{D^-,D^+} - \mu_{D^+,D^+}}. \tag{12}$$

El segundo método propuesto trata de completar el tipo de medidas consideradas para comparar distribuciones. Por un lado tenemos una medida basada en histogramas (Hellinger), una segunda basada en distancias (*Energy*) y se podría considerar una tercera opción que se basase en rankings, por ejemplo usando el criterio Cramér-von Mises (CvM). Después de analizar varias alternativas que utilizaban medidas derivadas de CvM [13], observamos que planteaban ciertos problemas en el contexto de la cuantificación binaria. El principal es que los errores en los rankings dados por un clasificador no son simétricos respecto a las clases: la clase positiva tiende a dar valores más altos, por ejemplo, al usar un clasificador probabilístico que devuelva $P(y = +1|x)$, lo que hace que los errores sean mayores y la medida sea sesgada hacia esa clase. Por ese motivo, finalmente decidimos usar el enfoque propuesto en [14] y crear el método CvMy que consiste en:

1. Calcular un ranking conjunto de los ejemplos disponibles en T, D^- y D^+,
2. Utilizar la ED para ajustar las distribuciones.

Es decir, aplicamos el mismo algoritmo que en el método EDy, pero en lugar de calcular la distancia entre las predicciones, calculamos la distancia entre los rankings de las predicciones. En este caso μ_{T,D^-} se calcularía como

$$\mu_{T,D^-} = \frac{1}{mn^-} \sum_{\boldsymbol{x}_j \in T} \sum_{\boldsymbol{x}_i \in D^-} ||r(h(\boldsymbol{x}_j)) - r(h(\boldsymbol{x}_i))||, \tag{13}$$

donde $r(h(\boldsymbol{x}))$ nos devuelve la posición en el ranking conjunto para todos los ejemplos disponibles, tanto del conjunto de entrenamiento como de la muestra de test a predecir. Luego la diferencia entre EDy y CvMy es simplemente que en un caso usamos las distancias entre las predicciones y en otro caso las diferencias entre los rankings de esas predicciones.

V. Experimentos

El objetivo de los experimentos[3] era comparar los métodos descritos en las secciones III y IV sobre varios conjuntos de datos. Se prestó especial atención a aquellos pares de métodos que, empleando la misma técnica, actúan sobre X ó y. Otro aspecto a analizar es la diferencia entre los métodos que utilizan distintas métricas para ajustar las distribuciones. Con el fin de tener una base de comparación, se incluyó el método AC. Para realizar los experimentos se emplearon 41 conjuntos de datos. Entre ellos se encuentran conjuntos binarios y otros que son versiones binarizadas de conjuntos originalmente multiclase. Así, por ejemplo, el conjunto balance.1 es un conjunto binario en el que la clase 1 original es la clase positiva, formando el resto la clase negativa.

Los métodos que necesitan un clasificador (AC, HDy, EDy y CvMy) fueron entrenados garantizando que usen exactamente los mismos clasificadores. Como algoritmo de clasificación se optó por utilizar *Random Forest* (RF) con salida probabilística para obtener modelos no lineales. La representación usada por el método HDy se generó mediante 8 *bins* por ser el valor que da mejores resultados en estudios previos [15]. Respecto a la hiperparametrización de RF, se llevó a cabo una búsqueda donde la profundidad variaba entre $[1, 5, 10, 15, 20, 25, 30]$, el número de árboles entre $[9, 18, 27, 36, 45, 54, 63]$, y el número mínimo de ejemplos para los nodos hoja entre $[1, 2, 4, 6, 8, 10]$. Dicha búsqueda se realizó optimizando la media geométrica mediante una validación cruzada de tres particiones de tal forma que se obtuvieran buenos clasificadores incluso cuando las clases no estuvieran balanceadas.

Todos los modelos se entrenaron sobre las mismas particiones de los conjuntos de datos en subconjuntos de entrenamiento y test, usando el $70\,\%$ de los datos para entrenar y el $30\,\%$ restante para testear el modelo, haciendo 20 repeticiones. Con cada partición de test se generaron a su vez 50 conjuntos aleatorios con diferente prevalencia mediante muestreo con reemplazamiento. La Tabla I presenta el error absoluto medio de las 1000 muestras totales de test, $\frac{1}{1000}\sum_{i=1}^{1000} |\hat{p}_i - p_i|$.

[3]Con el fin de facilitar la reproducibilidad de estos experimentos, los conjuntos de datos, el código y los resultados obtenidos están disponibles en http://github.com/albertorepo/analisis-cuantificacion.

Tabla I: Error absoluto medio para cada método sobre cada conjunto de datos

conjunto	AC	CvMy	EDX	EDy	HDX	HDy
acute.a	**0.037108**	0.063550	0.043139	0.045118	0.055545	0.038207
acute.b	**0.032788**	0.066146	0.036890	0.041999	0.041406	0.054910
balance.1	0.035783	0.033945	**0.023380**	0.030977	0.031761	0.029968
balance.3	0.036954	0.041549	0.031478	0.036346	0.038860	**0.028690**
breast-cancer	**0.016127**	0.041405	0.021433	0.019083	0.020661	0.016980
cmc.1	0.086332	0.078364	0.078307	0.076136	0.069633	**0.066261**
cmc.2	0.107941	0.119897	**0.067238**	0.115609	0.070623	0.108277
cmc.3	0.123313	0.102359	0.097244	0.094503	**0.089708**	0.099939
coil	**0.074547**	0.088761	0.091519	0.084081	0.152487	0.111920
ctg.1	**0.015548**	0.024147	0.029277	0.017414	0.039771	0.015466
ctg.2	0.029396	0.034666	0.033221	0.029420	0.035380	**0.027457**
ctg.3	0.036554	0.034602	0.037974	**0.028830**	0.072301	0.029866
default_credit	0.019715	0.022824	0.019998	0.020298	**0.019029**	0.022613
diabetes	0.072729	0.054864	0.059925	**0.053669**	0.079258	0.057511
german	0.087824	0.094541	**0.078250**	0.094154	0.100175	0.108456
haberman	0.243842	0.249899	**0.166582**	0.235816	0.263772	0.210012
ionosphere	**0.048585**	0.066306	0.060841	0.055436	0.055167	0.051017
iris.1	0.002824	0.093470	**0.016286**	0.019482	0.019413	0.036264
iris.2	0.055456	0.072105	0.083458	**0.048620**	0.050365	0.060695
iris.3	0.056763	0.066537	0.048433	0.046074	**0.040583**	0.075169
lettersH	0.020787	0.026393	0.027549	0.025537	0.024807	**0.016682**
mammographic	0.055443	0.049573	0.047413	0.044584	0.040516	**0.039722**
normtrans	0.126085	0.115678	**0.083582**	0.111233	0.184914	0.140455
normwine.1	0.044364	0.053990	**0.031585**	0.037656	0.036819	0.041440
normwine.2	0.050073	0.052678	**0.041247**	0.044185	0.067116	0.052616
normwine.3	0.043011	0.058478	**0.037641**	0.037738	0.058761	0.059418
pageblocks.5	0.042875	0.045982	0.056530	0.042546	0.116209	**0.037511**
phoneme	0.016554	0.019391	0.020960	0.015883	0.017325	**0.012115**
semeion.8	0.058507	0.058614	0.086534	0.058085	0.053439	**0.042278**
sonar	0.113939	0.108738	**0.103082**	0.107576	0.126948	0.110015
spambase	**0.008127**	0.015965	0.011639	0.010721	0.020363	0.008359
spectf	0.184524	0.162914	0.105445	0.150938	**0.088333**	0.134393
tictactoe	**0.048306**	0.061411	0.081694	0.056250	0.078846	0.048768
transfusion	0.107033	0.132669	**0.079578**	0.116017	0.162173	0.139883
wdbc	0.022888	0.030576	0.025309	0.016853	0.027982	**0.016830**
wine-quality-red	0.045189	0.037138	0.042694	**0.035102**	0.047469	0.037303
wine-quality-white	0.031770	0.030004	**0.028205**	0.028864	0.029644	0.025918
wine.1	0.037356	0.051096	0.034395	**0.033942**	0.036904	0.044647
wine.2	0.044871	0.056931	0.051078	**0.041804**	0.068114	0.051526
wine.3	0.037569	0.057219	0.036618	**0.035190**	0.048123	0.038674
yeast	0.060799	0.062919	**0.051120**	0.063225	0.060161	0.052552

El método ganador para cada conjunto se destaca en negrita. Puede verse que no hay un método que gane claramente a los demás. Tampoco se observan diferencias concluyentes entre las formas de estimar las distribuciones ni entre las medidas para ajustarlas. Entre los métodos propuestos, el algoritmo EDy es superior, siendo competitivo con los algoritmos previos, lo que parece no ocurrir en el caso del método CvMy.

Para analizar los resultados desde un punto de vista estadístico hemos aplicado análisis bayesianos en lugar de los tradicionales tests de contraste de la hipótesis nula ya que estamos de acuerdo con las desventajas de estos últimos apuntadas en [16]. Previo a la aplicación de estos tests, es necesario definir la *región de equivalencia práctica* (*rope*, por sus siglas en inglés): dos métodos se consideran prácticamente equivalentes si la diferencia dada una métrica no supera un cierto umbral. En nuestro caso, consideramos dos cuantificadores equivalentes si la diferencia en error absoluto es menor del $1\,\%$. La elección del error absoluto como medida de evaluación se debe a que es una métrica acotada entre 0 y 1 y por lo tanto, *rope* puede definirse como un porcentaje de variación dentro de ese intervalo. Para otras métricas como la KLD no sería trivial la elección de tal región, dado que su valor tiene un rango infinito, además de una difícil interpretación práctica. Una vez fijado el valor de *rope* es posible realizar el análisis para cada par de métodos tanto en cada conjunto individual (Tabla II) como para todos los conjuntos usando un test jerárquico [16] (Tabla III). En el primer caso podemos calcular la probabilidad de cada una de las tres posibilidades: que gane uno de los dos cuantificadores o que los resultados caigan en la zona de equivalencia. Cuando una de esas tres probabilidades es mayor que el $95\,\%$ consideramos que hay una diferencia significativa en favor de esa alternativa, y si no, lo marcamos como un conjunto *sin decisión*. Analizando la Tabla II vemos una ligera ventaja en favor de los métodos y frente a los métodos X, más clara en el caso de HDy frente a HDX que en la comparación EDy vs. EDX. En el caso de CvMy, es inferior a HDy y EDy, especialmente frente al segundo. En cambio EDy obtiene mejores resultados que el resto de métodos excepto HDy, pero la diferencia entre ambos es pequeña. Si analizamos los resultados a nivel global con el

Tabla II: Número de conjuntos para los que el test bayesiano decide que existe diferencia significativa ($\geq 95\,\%$)

Par ($m1$-$m2$)	$m1$	$rope$	$m2$	Sin decisión
EDX-AC	16	7	16	2
EDX-CvMy	17	11	7	6
EDX-EDy	9	16	12	4
EDX-HDX	10	15	7	9
EDX-HDy	10	15	14	2
EDy-AC	14	14	9	4
EDy-CvMy	11	24	0	6
EDy-HDX	15	11	9	6
EDy-HDy	8	13	10	10
HDX-AC	11	9	17	4
HDX-CvMy	16	9	13	3
HDX-HDy	6	13	19	3
HDy-AC	11	19	11	0
HDy-CvMy	15	15	6	5
AC-CvMy	16	11	9	5

Tabla III: Resultados del test bayesiano jerárquico

Par ($m1$-$m2$)	$P(m1 \gg m2)$	$P(rope)$	$P(m2 \gg m1)$
EDX-AC	0.537	0.000	0.463
EDX-CvMy	0.970	0.013	0.018
EDX-EDy	0.239	0.252	0.509
EDX-HDX	0.347	0.598	0.055
EDX-HDy	0.342	0.009	0.649
EDy-AC	0.523	0.202	0.275
EDy-CvMy	0.286	0.714	0.000
EDy-HDX	0.768	0.134	0.099
EDy-HDy	0.105	0.744	0.150
HDX-AC	0.162	0.002	0.837
HDX-CvMy	0.706	0.001	0.293
HDX-HDy	0.022	0.005	0.973
HDy-AC	0.451	0.072	0.477
HDy-CvMy	0.971	0.005	0.024
AC-CvMy	0.814	0.005	0.180

test jerárquico (Tabla III) vemos que hay pocas diferencias significativas (HDy vs. CvMy, HDy vs. HDX y EDy vs. CvMy). Se observa de nuevo la ligera ventaja de los métodos y, EDy y HDy, frente a EDX y HDX. Estos datos son un resumen de la distribución de las diferencias entre cada par de métodos. La forma de la distribución puede verse mediante un gráfico simplex (Figura 2). Por ejemplo, es interesante ver que en la comparación EDy-CvMy la nube de puntos está muy alejada del vértice de CvMy, a pesar de que la diferencia no resulte significativa, mostrando una superioridad clara.

VI. Conclusiones

Este trabajo tenía como principal objetivo realizar un análisis comparativo entre diferentes algoritmos de cuantificación basados en ajuste de distribuciones. Los resultados experimentales obtenidos muestran que los métodos basados en las y's, es decir, en el uso de clasificadores, no son en general peores, obteniendo mejores estimaciones en ciertas ocasiones. En cuanto a los métodos propuestos, EDy se configura como una buena alternativa, pero no así el algoritmo propuesto basado en ranking, CvMy. Como futura línea de investigación, se plantea la aplicación de algún algoritmo de selección de características en los métodos X para tratar de obtener un subconjunto con los mejores atributos. Además, sería interesante realizar una búsqueda de características en los problemas

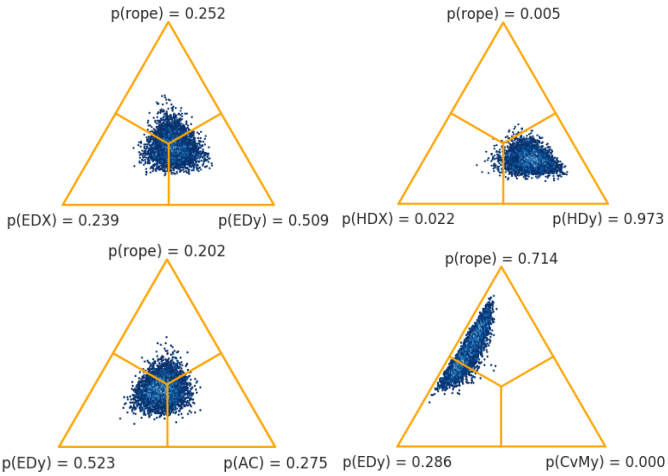


Figura 2: Gráficos simplex de la comparación mediante el test jerárquico bayesiano de varios pares de métodos

(alta dimensionalidad de entrada, presencia de ruido, etc.), que puedan implicar un mejor comportamiento en cada método.

Referencias

[1] G. Forman, "Quantifying counts and costs via classification," *Data Mining and Knowledge Discovery*, vol. 17, no. 2, pp. 164–206, 2008.

[2] P. González, J. Díez, N. Chawla, and J. J. del Coz, "Why is quantification an interesting learning problem?" *Progress in Artificial Intelligence*, pp. 1–6, 2016.

[3] V. González-Castro, R. Alaiz-Rodríguez, and E. Alegre, "Class distribution estimation based on the hellinger distance," *Information Sciences*, vol. 218, pp. 146–164, 2013.

[4] M. Sugiyama, T. Kanamori, T. Suzuki, M. C. du Plessis, S. Liu, and I. Takeuchi, "Density-difference estimation," *Neural Computation*, vol. 25, no. 10, pp. 2734–2775, 2013.

[5] A. Margolis, "A literature review of domain adaptation with unlabeled data," University of Washington, Tech. Rep., 2011.

[6] J. Barranquero, P. González, J. Díez, and J. J. del Coz, "On the study of nearest neighbor algorithms for prevalence estimation in binary problems," *Pattern Recognition*, vol. 46, no. 2, pp. 472–482, 2013.

[7] P. González, A. Castaño, N. V. Chawla, and J. J. del Coz, "A review on quantification learning," *ACM Computing Surveys*, vol. 50, no. 5, pp. 74:1–74:40, 2017.

[8] J. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *Pattern Recognition*, vol. 45, no. 1, pp. 521–530, 2012.

[9] A. Firat, "Unified framework for quantification," *arXiv preprint arXiv:1606.00868*, 2016.

[10] H. Daume III and D. Marcu, "Domain adaptation for statistical classifiers," *JAIR*, vol. 26, pp. 101–126, 2006.

[11] A. Iyer, S. Nath, and S. Sarawagi, "Maximum mean discrepancy for class ratio estimation: Convergence bounds and kernel selection." in *ICML*, 2014, pp. 530–538.

[12] H. Kawakubo, M. C. Du Plessis, and M. Sugiyama, "Computationally efficient class-prior estimation under class balance change using energy distance," *IEICE Tran. on Inf. and Sys.*, vol. 99, pp. 176–186, 2016.

[13] L. Morán-Fernández, V. Bolón-Canedo, and A. Alonso-Betanzos, "A distributed approach for classification using distance metrics," in *ESANN*, 2017.

[14] J. Curry, X. Dang, and H. Sang, "A rank-based Cramér-von-Mises-type test for two samples," *arXiv preprint arXiv:1802.06332*, 2018.

[15] P. Pérez-Gállego, J. R. Quevedo, and J. J. del Coz, "Using ensembles for problems with characterizable changes in data distribution: A case study on quantification," *Information Fusion*, vol. 34, pp. 87–100, 2017.

[16] A. Benavoli, G. Corani, J. Demšar, and M. Zaffalon, "Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis," *Journal of Machine Learning Research*, vol. 18, no. 77, pp. 1–36, 2017.