Universidad Nacional de Educación a Distancia
(UNED)
Escuela Técnica Superior de Ingeniería Informática
Máster en I.A. avanzada: fundamentos, métodos y aplicaciones.
Trabajo Fin de Master

Cuantificación y Ensembles.

Autor:
Miguel Rodríguez Valles
Tutor:
Enrique Amigó Cabrera
Convocatoria:
Septiembre
Año:
2022

Índice

1. Introducción

En este trabajo se pretende analizar la aplicación de las técnicas de *Ensemble* a los problemas de *Cuantificación*.

La *Cuantificación* es una de las tareas que recientemente ha adquirido cierto protagonismo dentro de las diferentes problemáticas que aborda el denominado aprendizaje automático (o machine learning). En ella se busca determinar la distribución de una colección de clases en un conjunto sin etiquetar; con este fin se ha desarrollado una colección de técnicas específicas que se aglutinan dentro del término *Cuantificación*.

Se puede considerar un pariente cercano, o lejano según se mire, de la *Clasificación*, si bien existen diferencias que son necesarias poner de manifiesto. En la tarea de la *Clasificación* se obtiene un algoritmo, lo más preciso posible[1], h^C, que asigna una clase, c_i, a cada observación, x_k, del conjunto sin etiquetar, $\mathcal{T}$:

$$h^C \quad | \, h^C(x_k) \longrightarrow c_i \quad x_k \in \mathcal{T}$$

La *Cuantificación* busca determinar la distribución de probabilidad de dichas categorías, c_i, en el conjunto sin etiquetar $\mathcal{T}$ a través de un algoritmo h^Q:

$$h^Q \quad | \, h^Q(\mathcal{T}) \longrightarrow (P(c_1), ..., P(c_m))$$

En este planteamiento se evidencia la primera diferencia relevante entre ambas tareas, mientras que la *Clasificación* pretende la correcta clasificación de cada una de las observaciones de $\mathcal{T}$, la *Cuantificación* se preocupa por la muestra completa y no por cada observación. Esta diferencia tendrá implicaciones, que serán descritas a lo largo del documento, tanto en la utilización de clasificadores para la *Cuantificación* como en las medidas de error que hay que definir en esta tarea.

Otra diferencia relevante a la que se enfrenta la *Cuantificación* frente a la *Clasificación* es la hipótesis que descansa sobre el P.G.D.[2] que genera los conjuntos de datos de entrenamiento y test. Mientras que la *Clasificación* se basa en la hipótesis I.I.D.[3], que implica que tanto el conjunto de entrenamiento como el conjunto test provengan de la misma distribución, la *Cuantificación* no asume que los datos de entrenamiento y test provienen de la misma distribución. Esta discrepancia entre la distribución en entrenamiento y test es una de las razones por las que la aplicación directa de los clasificadores no arroja resultados resultados satisfactorios cuando se aborda la *Cuantificación*.

Las diferencias apuntadas sugieren que la *Cuantificación* desarrolle un acervo propio de técnicas que permita soslayar los incovenientes anteriormente mencionados. A lo largo del documento éstas técnicas serán abordadas durante los

[1] Y no entraremos a especificar el sentido concreto de precisión.
[2] Proceso Generador de Datos
[3] Independencia e Identicamente Distribuidas.

primeros epígrafes del presente documento.

El otro aspecto importante del documento gira en torno a los *Ensemble*; esta clase de algoritmos se caracteriza por la combinación bajo algún criterio, al cual denotaremos por f, de una colección de modelos base $\{h_1, ..., h_N\}$, dando lugar a un *macro-modelo* H:

$$H = f(h_1, ..., h_N)$$

Esta metodología, que tiene su mayor predicamento en el ámbito de la clasificación, ha mostrado éxito en la mayor parte de las comparativas con las técnicas tradicionales. Este éxito reside, entre otros aspectos, en la diversidad de modelos base que la propia técnica genera; es esta diversidad la que pudiera servir de fulcro en el que apoyarse para paliar la ausencia de IID en un problema *Cuantificación*. Por tanto hemos de valorar si ante un problema de *Cuantificación* la utilización de *Ensembles* puede presentar una alternativa a las técnicas desarrolladas hasta el momento.

La potencial complementariedad que pudieran tener la *Cuantificación* y los métodos *Ensemble* suscita la cuestión de hasta qué punto la aplicación de este tipo de metodologías, *Ensemble*, supone una mejoría frente a las técnicas propias con las que nos enfrentamos al problema de la *Cuantificación*. Para responder a esta cuestión se ha elaborado este documento cuya organización pretende estructurar de forma lógica una respuesta a la pregunta planteada. Se comenzará con la definición del problema así como de los principales enfoques actualmente seguidos para abordarla con éxito. Con posterioridad se incorpora un bosquejo tanto del fundamento como de los métodos *Ensemble* que existen en la actualidad para con posterioridad analizar la aplicabilidad de éstos sobre aquella. Este análisis se llevará a cabo tanto desde un plano bibliográfico, revisando la literatura actual existente, como empírico, realizando una comparativa de resultados sobre diversos conjuntos de datos de las técnicas tradicionales de *Cuantificación* y las técnicas *Ensemble*. Por último daremos respuesta a la cuestión planteada sobre la base de los resultados obtenidos en los experimentos llevados a cabo.

2. Fundamentos teóricos

2.1. Cuantificación

2.1.1. Planteamiento y formalización del problema

Tal como hemos adelantado en la introducción el problema de *Cuantificación* se enfrenta a la determinación de la distribución de probabilidad, $\{P_i\}_{i=1}^{m}$, de un conjunto de clases, $\mathcal{C} = \{c_1, ..., c_m\}$, sobre una conjunto de observaciones sin etiquetar, $\mathcal{T}$:

$$\mathcal{T} = \{x_1, \ldots, x_M : x_i \in \mathcal{X}\}$$

siendo $\mathcal{X}$ el espacio donde se definen los rasgos que caracterizan las observaciones.

Es una labor supervisada y, por tanto, se dispone de un conjunto de entrenamiento etiquetado, $\mathcal{S}$:

$$\mathcal{S} = \{(x_1, y_1), \ldots, (x_N, y_N) : (x_i, y_i) \in \mathcal{X} \times \mathcal{C}\}$$

Este conjunto $\mathcal{S}$ será utilizado para obtener el algoritmo de *Cuantificación*, h_S^Q, que dará respuesta a la tarea:

$$h_S^Q : \mathcal{P}(\mathcal{X}) \longrightarrow [0,1]^C$$
$$h_S^Q(\mathcal{T}) = (P(\hat{c_1}), ..., P(\hat{c_m}))$$

donde hemos denotado por $\mathcal{P}(\mathcal{X})$ el conjunto de muestras de $(\mathcal{X})$. Con carácter general usaremos el supraíndice $\hat{}$ para indicar que se trata de una estimación. En caso de disponer de un clasificador, entrenado sobre la muestra de entrenamiento, se le denotará por h_S^C, y estará definido de la siguiente manera:

$$h_S^C : \mathcal{X} \longrightarrow \mathcal{C}$$
$$h_S^C(x) = c$$

De cara a manejar una notación más sencilla no se mostrará la dependencia explícita de la muestra de entrenamiento, el subíndice, tanto en el cuantificador como en el clasificador siempre que no sea estrictamente necesario.

Denotaremos por L^C y L^Q las respectivas funciones de pérdida de *Clasificación* y *Cuantificación*. A la familia de funciones donde se ubicará tanto el clasificador como el cuantificador le reservaremos la notación $\mathcal{H}$, de esta manera podemos definir las funciones de pérdida de la siguiente manera:

$$L^C : \mathcal{X} \times \mathcal{H} \longrightarrow \mathbb{R}$$
$$L(\mathcal{S}, h_S^C)$$

$$L^Q : \mathcal{P}(\mathcal{X}) \times \mathcal{H} \longrightarrow \mathbb{R}$$
$$L(\mathcal{S}, h_S^Q)$$

Se utilizará la letra P para denotar la probabilidad añadiéndose como argumento la clase a la que se hace referencia; cuando se esté ante una estimación se usará el acento circunflejo para distinguirlo del valor real. Si fuese necesario indicar el conjunto sobre el que se calcula, éste se añadirá como subíndice. De acuerdo a lo anterior las principales notaciones utilizadas a lo largo del documento serán las siguientes:

- $P_{\mathcal{T}}(c_j)$: Probabilidad real de la clase j-ésima en la muestra $\mathcal{T}$.
- $P_{\mathcal{T}}(\hat{c_j})$: Estimación de la probabilidad de la clase j-ésima en la muestra $\mathcal{T}$.
- $P_{\mathcal{T}}(\hat{c_j}|c_i)$: Probabilidad estimada de la clase j-ésima en la muestra $\mathcal{T}$ condicionada a que su clase real es la i-ésima.

2.1.2. Clasificación y Cuantificación

Clasificación y *Cuantificación* son tareas que están estrechamente relacionadas entre sí aunque persigan objetivos diferentes. Ambas se tratan de tareas supervisadas en las que el objetivo es la asignación de clases sobre un conjunto sin etiquetar, si bien a niveles diferentes: la *Clasificación* asigna las etiquetas a nivel observación mientras que la *Cuantificación* lo realiza a nivel conjunto. De cara a mostrar los aspectos más relevantes en los que difieren ambas tareas esquematizaremos los marcos generales en los que se desenvuelven éstas.

En el gráfico inferior se ha esbozado, basándonos en el esquema de utilizado por Y. Abu-Mustada et el en [AML12], la tarea de *Clasificación*, mostrando los principales elementos que intervienen tanto en la fase de entrenamiento como en la de aplicación.

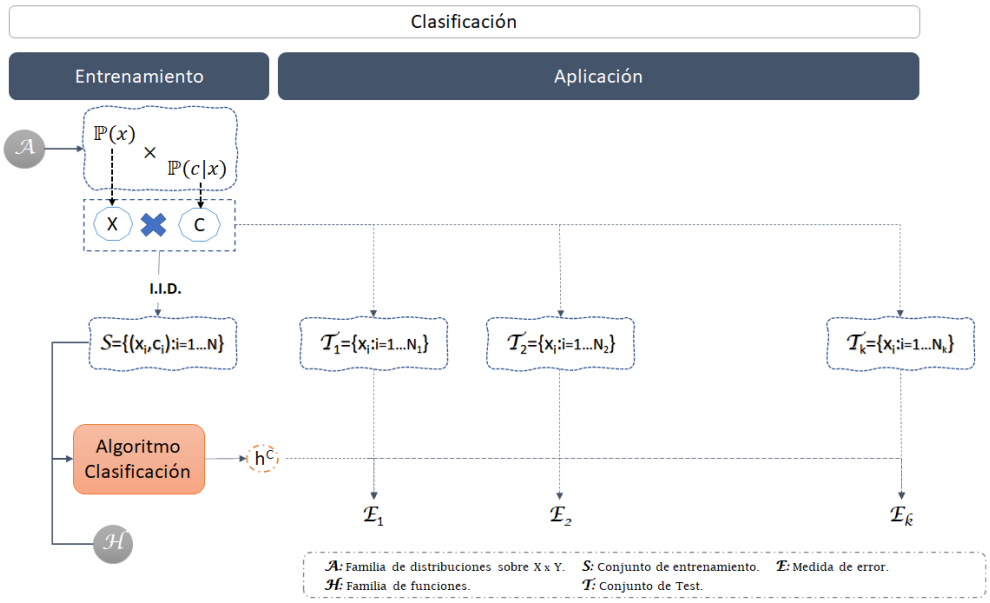

Figura 1: Tarea Clasificación.

En la fase de entrenamiento se dispone del conjunto $\mathcal{S}$ que está constituido por observaciones generadas por distribuciones de probabilidad desconocidas pero que se asumen fijas, a saber:

- $\mathbb{P}(x)$: distribución de probabilidad de los rasgos.
- $\mathbb{P}(c|x)$: distribución de probabilidad de la clase condicionada a un rasgo.

Las observaciones son generadas de forma independiente de acuerdo a estas distribuciones. A este conjunto de hipótesis que caracterizan la generación de datos se le denomina I.I.D. (Independientes e Identicamente Distribuidas).

La familia de funciones $\mathcal{H}$ está fijada de antemano y representa el conjunto de potenciales clasificadores sobre los que buscará el Algoritmo de Clasficación el óptimo[4], h^C. Este clasificador será evaluado durante la etapa de aplicación en diferentes conjuntos de test $\mathcal{T}$ generados con la misma distribución de rasgos, $\mathbb{P}(x)$, y que a posteriori dispondrán de sus correspondientes etiquetas a través de la distribución condicionada, $\mathbb{P}(y|x)$. Este último aspecto permite medir los errores cometidos, $\mathcal{E}$, en cada una de las muestras.

En el caso de la *Cuantificación* un diagrama equivalente sería el siguiente:

[4]El denominado Algoritmo de Clasificación engloba a la función de error que será optimizada y arrojará el clasificador óptimo.

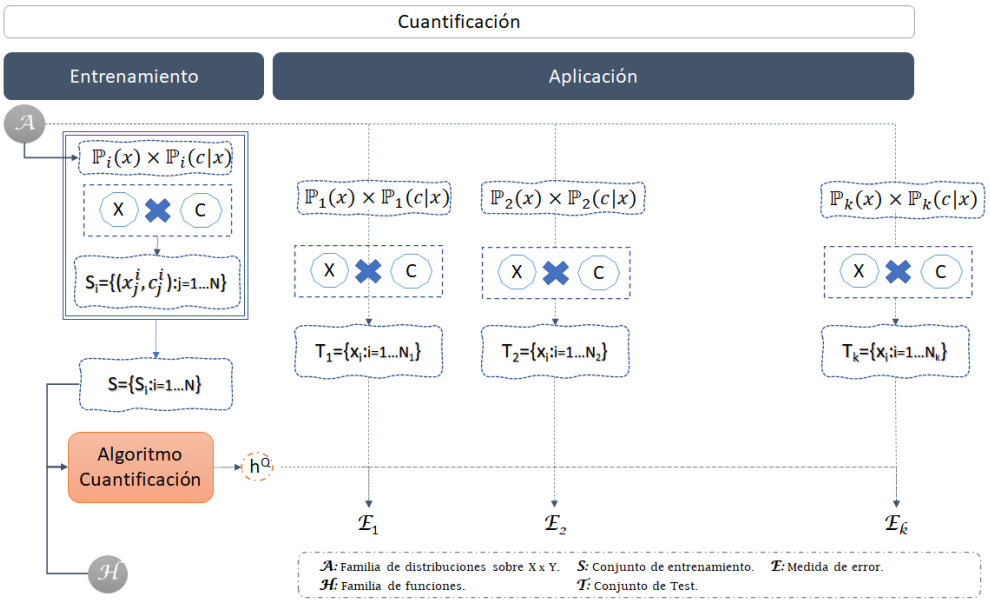

Figura 2: Tarea Cuantificación.

El esquema es similar al anterior, sin embargo, difiere en dos aspectos importantes:

- **Hipótesis sobre la distribución generadora de datos**.
  La *Cuantificación* no mantiene la hipótesis de una única distribución generadora de datos sino que es la diversidad de éstas la que se establece. A raíz de esto el conjunto de entrenamiento, $\mathcal{S}$, en la *Cuantificación* no estaría conformado únicamente por una única muestra extraída de una distribución sino que debería recoger una cantidad suficiente de éstas obtenidas de tantas distribuciones. Este aspecto no es exclusivo de la fase de entrenamiento sino que se mantiene en la fase de aplicación. Con todo ello la hipótesis de I.I.D. establecida en la *Clasificación* no es propia de la *Cuantificación*.

- **Algoritmo de selección del Clasificador**.
  A raíz de la anterior singularidad los algoritmos tradicionales desarrollados en las tareas de *Clasificación* no son directamente aplicables a la *Cuantificación* ya que las funciones de pérdida no son iguales sino que difieren en como miden el error, mientras que en la *Clasificación* es relevante el volumen de errores en la *Cuantificación* es más relevante la distribución de éste. Esto conllevará que los algoritmos de clasificación deban ser modificados para atender a esta nueva tarea.

Estas dos diferencias han sido puestas de manifiesto en la mayor parte de las

fuentes bibliográficas consultadas tal como G. Forman en [For05] o Sebastiani en [Seb18b] y son lo suficientemente relevantes para impedir la reducción de la *Cuantificación* a la *Clasificación*, por ello merece la pena profundizar en las implicaciones que tienen cada una de ellas en la *Cuantificación* tal como se presentará a continuación.

- **Irrelevancia de la hipótesis I.I.D.: Shift Distribution**.

  Hemos visto que la *Clasificación* se asienta sobre la hipótesis I.I.D., es decir, tanto el conjunto de entrenamiento como los diversos conjuntos de test/aplicación están conformados por observaciones generadas de forma Independiente e Identicamente Distribuidas, es decir, se verifica que:

$$\begin{cases} \mathbb{P}_S(x) = \mathbb{P}_T(x) \ \forall x \\ \mathbb{P}_S(c|x) = \mathbb{P}_T(c|x) \ \forall x, c \end{cases}$$

  y por tanto:

$$\sum_{x \in \mathcal{X}} \mathbb{P}_S(c|x)\mathbb{P}_S(x) = \sum_{x \in \mathcal{X}} \mathbb{P}_T(c|x)\mathbb{P}_T(x)$$

  es decir:

$$P_{\mathcal{S}}(c) = P_{\mathcal{T}}(c) \ \forall c \in \mathcal{C}$$

  La verificación de esta hipótesis en el ámbito de la *Cuantificación* permitiría la resolución del problema, al menos desde un marco teórico, mediante la aplicación de la *Ley de los Grandes Números*, la cual nos garantizará la convergencia de las frecuencias observadas a las probabilidades de cada clase:

$$P(\hat{c}_j) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{y_i = c_j\} \longrightarrow \mathbb{E}[\mathbb{I}\{y_i = c_j\}] = P_j, \quad \forall \ 1 \leq j \leq m$$

  Sin embargo, en las situaciones donde surge la *Cuantificación* el mantenimiento de la hipótesis I.I.D. no es razonable, ya que esa variación de la distribución es intrínseca a los casos donde surge la tarea. Este fenómeno recibe la denominación de *Shift Distribution*.

  Uno de los problemas más habituales a los que se enfrenta la *Cuantificación* es en la determinación del sentimiento en redes sociales (por ejemplo en Twitter) donde los cambios de parecer son rápidos y notables [5].

$$\begin{cases} \text{Clasificacion} \Rightarrow \mathbb{P}_{Train}(x, c) = \mathbb{P}_{Test}(x, c) = \mathbb{P}_{Aplicacion}(x, c) \\ \text{Cuantificacion} \Rightarrow \mathbb{P}_{Train}(x, c) \neq \mathbb{P}_{Test}(x, c) \neq \mathbb{P}_{Aplicacion}(x, c) \end{cases}$$

[5] Situaciones como la apuntada son recogidas en algunos de los trabajos aplicados de la *Cuantificación* como [Mil+15], [GS15].

Esta diferencia dificulta que la aplicación directa de un clasificador arroje resultados satisfactorios en un problema de *Cuantificación*. Este hecho lo formalizó Forman en [For07] mediante la siguiente proposición.

Proposition *Sea $\mathcal{S} \subseteq \mathcal{X} \times \mathcal{Y}$ el conjunto de entrenamiento de un problema de clasificación binario en el que las ratio de verdaderos positivos (TPR) y falsos positivos (FPR) son constantes*[6] *y sea p la frecuencia de casos positivos en esta muestra. Entonces si h^C es un clasificador imperfecto, $TPR \neq FPR$, entrenado en este conjunto que verifica:*

$$h^C(\mathcal{S}) = p$$

ante cualquier variación de la probabilidad de positivos, p', en el conjunto test, $\mathcal{T}$, el clasificador no se mantendrá calibrado:

$$h^C(\mathcal{T}) \neq p'$$

Una demostración alternativa a la realizada por Forman puede encontrarse en el anexo.

El siguiente ejemplo, que hace uso de la proposición anterior, nos permite visualizar la problematica; así, supongamos un clasificador h que se enfrenta a un tarea de *Cuantificación* tras haber sido entrenado en un conjunto $\mathcal{S}$ con una tasa de positivos p =0.4. El clasificador consigue en $\mathcal{S}$ predecir exactamente p, sin embargo fallará a la hora de predecir nuevas tasas en muestras de test con diferentes niveles de prevalencia, tal como se puede apreciar en la gráfica inferior:

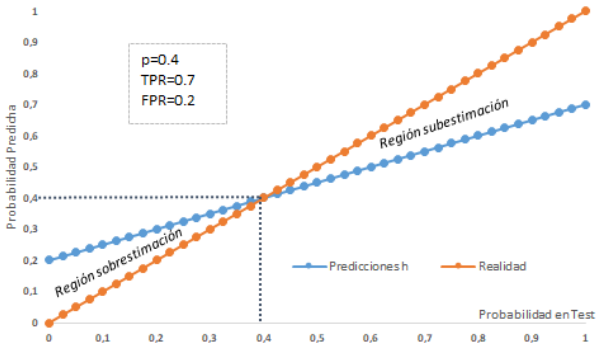

Figura 3: Relación Cuantificación y Clasificación.

Es por ello necesario desarrollar nuevos métodos para abordar este tipo de problemas.

[6] Un aspecto que puede resultar sospechoso en la proposición anterior es la exigencia de la constancia de las ratios TPR y FPR, a pesar de la exigencia de esta condición es una hipótesis habitualmente asumida en todas las aproximaciones a la resolución de los problemas de *Cuantificación* como veremos con posterioridad en la exposición de los métodos de resolución.

- **Medición del error**.

  Otro de lo aspectos en los que difieren ambas tareas es en la definición, y por tanto la medición, del error. Esto es una consecuencia directa de los diferentes objetivos que persiguen. La *Clasificación* persigue etiquetar correctamente una observación cualquiera, y consecuentemente cada error individual cuenta (el volumen de éstos es lo relevante), la *Cuantificación* busca predecir correctamente la distribución de categorías en una muestra cualquiera, y por tanto los errores individuales no son relevantes sino su distrubución. Esta diferencias tendrán diversas implicaciones, así:

  - La diferencia entre las dos funciones de error ($L^Q \neq L^C$) conlleva que generalmente la optimalidad alcanzada en una tarea no implique su consecución en la otra. Para analizar este punto supongamos un problema binario, $c \in \{-1, +1\}$, en el que se desea llevar a cabo tanto la tarea de *Cuantificación* como de *Clasificación*. Las condiciones de optimalidad que deberan satisfacer los algoritmos son las siguientes:
    - *Problema de Clasificación.* En la tarea de *Clasificación* un algoritmo h^C minimizará el error empírico de clasificación[7], L^C, sobre la muestra de construcción $\mathcal{S}$:

$$L^C(h;\mathcal{S}) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(c_i \neq h(x_i)) = \frac{1}{N}\sum_{i|c_i=1}\mathbb{I}(h(x_i) \neq 1) + \\ + \frac{1}{N}\sum_{j|c_i=-1}\mathbb{I}(h(x_i) \neq -1) = \frac{1}{N}[FP(h) + FN(h)]$$

      donde FP y FN representan el volumen de Falsos Positivos (FP) y Falsos Negativos (FN). El clasificador óptimo h^C será aquel que minimice esta última expresión:

$$h^C = \arg\min_h\{FP(h) + FN(h)\}$$

    - *Problema de Cuantificación.* En el caso de la *Cuantificación* binaria el error cometido, L^Q, puede expresarse mediante la diferencia entre lo observado, p, y lo predicho con un modelo h, $\hat{p}(h)$, así:

$$L^Q(h;\mathcal{S}) = |\hat{p}(h) - p| = |\frac{TP(h) + FP(h)}{N} - \frac{TP(h) + FN(h)}{N}| = \\ = \frac{|FP(h) - FN(h)|}{N}$$

      donde TP son los Verdaderos Positivos. Por tanto el cuantificador h^Q óptimo será aquel que minimice dicha expresión:

[7] A estos efectos asumiremos que $L_C(x, y) = 1 - \delta_{xy}$ siendo δ la delta de Kronecker.

$$h^Q = \arg\min_h \{|FP(h) - FN(h)|\}$$

Por tanto:

$$L^Q(h^Q; \mathcal{S}) \leq L^Q(h^C; \mathcal{S})$$

y la igualdad sólo se dará cuando los dos errores sean nulos:

$$L^Q(h^Q; \mathcal{S}) = L^Q(h^C; \mathcal{S}) \iff FN = 0 \text{ y } FP = 0.$$

es decir sólo en el caso de disponer de un clasificador perfecto, FP=0 y FN=0, su aplicación al problema de *Cuantificación* nos aseguraría verificar la condición de optimalidad; en el resto de situaciones, las más comunes, la aplicación del clasificador a un tarea de *Cuantificación* no es óptima.

- Otra de las consecuencias es que no existe una relación directa entre ser un buen clasificador y ser un buen cuantificador, es decir que dados clasificadores h_1^C y h_2^C:

$$L^C(h_1^C; \mathcal{S}) \leq L^C(h_2^C; \mathcal{S}) \nRightarrow L^Q(h_1^C; \mathcal{S}) \leq L^Q(h_2^C; \mathcal{S})$$

Un simple ejemplo nos permite comprobar la veracidad de la afirmación realizada:

Clasificador $h_{C,1}$

Matriz de Confusión		*Predicho*		*Total Observado*
		1	-1	
Obs	1	50	20	70
	-1	80	50	130
Total Predicho		130	70	200

Error Clasificación	0,5
$P_{obs}(1)$	0,35
$P_{pred}(1)$	0,65
Error Cuantificación	0,3

Clasificador $h_{C,2}$

Matriz de Confusión		*Predicho*		*Total Observado*
		1	-1	
Obs	1	10	60	70
	-1	80	50	130
Total Predicho		90	110	200

Error Clasificación	0,7
$P_{obs}(1)$	0,35
$P_{pred}(1)$	0,45
Error Cuantificación	0,1

Figura 4: Relación Cuantificación y Clasificación.

Por tanto la utilización de un buen clasificador entrenado para una tarea de *Clasificación* no garantizará que su aplicación a una tarea de *Cuantificación* sea adecuada.

2.1.3. Aplicaciones y variedades de la Cuantificación

Son diversos los ámbitos donde la *Cuantificación* tiene o puede tener cabida, basta buscar predicciones de agregados bajo condiciones de bastante volatilidad[8]. A continuación se muestran los ámbitos de aplicación más destacados:

[8]Utilizamos este calificativo para expresar el hecho de que requerimos que no se verifique la condición IID.

- **Análisis del sentimiento**. Mucho del interés por la *Cuantificación* surgió en el ámbito del análisis del sentimiento en redes sociales donde la utilización de esta técnica se ajusta perfectamente a las características mencionadas con anterioridad. En primer lugar su objetivo se adapta a cuestiones de interés como: *¿cuál es el sentimiento de un conjunto de tweets?*, *¿cuál es el tweet más popular en un determinado momento temporal?*, *¿cuál es la reacción de los usuarios a una determinada acción?*...Por otro lado las redes sociales son un ejemplo palmario de lo que no es un entorno IID; los cambios de opinión, el flujo constante de nueva información, entre otros, desautorizan asumir como cierta la hipótesis IID. Dentro la literatura revisada esta aplicación es la que más interés ha suscitado, así Sebastiani ha dedicado parte de su investigación en la *Cuantificación* a esta aplicación tal como atestiguan [Mil+15], [GS15], [EFS18] o [Seb18c]. Otros autores como Vilares et al. en [Vil+16], Karpov en [KLV18], Amati et al. en [ABB] o [Ayy+22] también abordan esta aplicación.

  Dentro de esta temática también podríamos englobar las aplicaciones en otros campos que recogen inquietudes similares. La valoración de la opinión de un candidato en el ámbito político como ha llevado cabo Hopkins y King en [HK10] o la percepción de un determinado producto en un estudios de mercado son ejemplos de aplicaciones de las técnicas de *Cuantificación*.

- **Epidemiología**. Dentro de la epidemiología la *Cuantificación* lleva tiempo siendo utilizada principalmente para la determinación de la prevalencia de enfermedades o para la estimación de las tasa de mortalidad de determinadas afecciones. Si bien en este ámbito no es habitual denominar como *Cuantificación* a los métodos que utilizan para conseguir sus fines, éstos son análogos a los que se aplican en otros usos más canónicos. Trabajos como el de Goldstein et al. en [Gol+11] aplican, sin mencionarlo expresamente, métodos de *Cuantificación* con el fin de estimar las prevalencias de determinadas enfermedades respiratorias.

  En lo que respecta a la estimación de las tasa de mortalidad, la *Cuantificación* ha sido utilizada en los procedimientos de Autopsia Verbal; éstos pretenden estimar la distribución de las causas de muerte en poblaciones en las que no es posible disponer de las certificaciones médicas de la causa de la defunción tras un examen forense sino de informes escritos sobre los síntomas. En este ámbito destaca el trabajo pionero de King y Lu en [KL08].

- **Finanzas y Economía**. También son diversas las aplicaciones en este campo de la *Cuantificación*. Forman en [For07] va más allá del enfoque clásico de la *Cuantificación* para desarrollar la *Cuantificación de Costes*, en la que en vez que estimar la distribución de probabilidad de una determinada clase de elementos estima la distribución de costes. Tasche en [Tas16] aborda la aplicación de la *Cuantificación* en el ámbito de la medición del riesgo de crédito, lo cual puede ser de utilidad a la hora de

calibrar las probabilidades de impago que arrojan los Scorings de calidad crediticia.

Las estimaciones de ciertos agregados macroeconómicos como pueden ser las tasa de paro en función de diversos ejes tales como la región geográfica, sexo, tramo de edad...son un ámbito en el que tendría cabida la aplicación de la *Cuantificación* si bien aun no ha sido explorado.

- **Problemas de clasificación**. La *Clasificación* puede beneficiarse de la *Cuantificación* en aquellos casos en los que es necesaria la disposición de la probalidad apriori de la clase. En esta situación se encuentran algoritmos como Naive Bayes o el análisis discriminante. En estos algoritmos la asignación de la clase a un elemento x se realiza de acuerdo a la siguiente expresión:

$$c = \arg\max_{c_i} \mathbb{P}(c_i|x) = \arg\max_{c_i} \{\mathbb{P}(x|c_i)\mathbb{P}(c_i)\}$$

Es decir se descompone el problema en dos partes, por un lado la determinación de la probabilidades condicionadas a cada clase, $\mathbb{P}(x|c_i)$, y por otro la determinación de las probabilidades a priori de las clases $\mathbb{P}(c_i)$. Es en este último punto donde la utilización de la *Cuantificación* permite mejorar el desempeño de este tipo de clasificadores.

La revisión de las aplicaciones de la *Cuantificación* nos muestra que a pesar de la univocidad de su definición existen diversas variedades bajo el término genérico. Para caracterizar estas tipologías tomaremos como criterio de análisis la naturaleza del *target*. Bajo este criterio las combinaciones posibles son las siguientes:

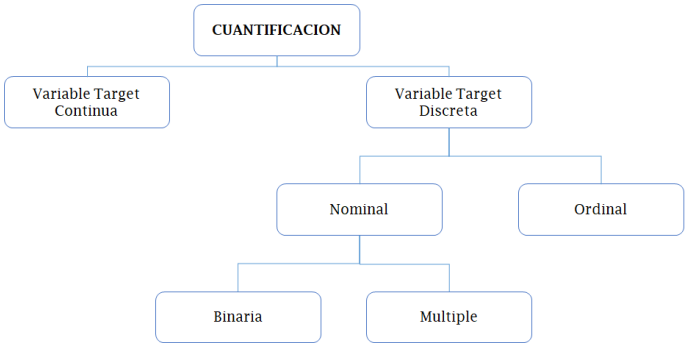

Figura 5: Tipologías de Cuantificación según target.

1. *Variable target continua*. Aunque en la definición presentada de *Cuantificación* hacíamos referencia a una distribución discreta de categorías no hay nada que impida la extensión de la problemática a una variable continua, es decir, en este tipo de probemos estaremos interesados en predecir

la distribución de una variable continua. Hasta el momento no ha habido mucho desarrollo en este ámbito siendo el trabajo más relevante el de Bella et al.[Bel+13], los cuales bautizan esta aplicación como *Cuantificación para la Regresión*. En general las ideas presentadas adaptan, como veremos con posterioridad, los métodos desarrollados en la Cuantificación discreta al mundo de la regresión.

2. *Variable target nominal multiple*. Cuando se dispone de más de dos clases y no existe ningún orden sobre ellas estamos ante este tipo de problemas. Su utilización es habitual en los problemas de Análisis del Sentimiento que hemos mencionado al inicio del epígrafe.

3. *Variable target nominal binaria*. Sin duda es el campo más explorado, y explotado, en la *Cuantificación* tanto por su utilidad como por su sencillez. Será esta tipología de *Cuantificación* la que usaremos para el análisis empírico que llevaremos a cabo en la última parte del documento.

4. *Variable target ordinal*. La última tipología de *Cuantificación* permite abordar la presencia de un orden parcial entre las clases. Aunque la *Cuantificación* per se es insensible a dicho orden la presencia de esta ordenación puede ser incorporada en el proceso de *Cuantificación*. Dentro de esta tipología cabe reseñar los siguiente documentos que abordan esta problemática A. Esuli en [Esu16], G. San Martino et al.en [DGS16] y T. Sakai en [Sak21].

2.1.4. Taxonomía de las técnicas

Una vez mostrado el entendimiento del problema se procederá a abordar en el resto del epígrafe la descripción de las principales técnicas que actualmente están en uso así como las principales medidas de desempeño y el diseño experimental seguido ya que ambos difieren de los habitualmente llevados a cabo en problemas de *Clasificación*.

Sebastiani en [Seb18b] elabora una taxonomía de las diferentes técnicas existentes tomando como criterio principal la dependencia de ésta de un clasificador, esta clasificación ha sido tomada en la mayor parte de artículos o estudios sobre el estado de arte de la cuestión como en [Gon+17] y será la que seguiremos. Según este criterio se identifican dos familias:

- **Agregativas**: se engloban bajo esta denominación todas aquellas técnicas que parten de un clasificador base para realizar la estimación de la distribución de las categorías a través de ciertas transformaciones o manipulaciones del mismo.

- **No Agregativas**: al contrario que en las anteriores no se requiere la disposición ni desarrollo de ningún clasificador, haciendo uso únicamente de los rasgos disponibles para llevar a cabo la estimación.

Cada una de estas familias dispone a su vez de diferentes variantes, tal como se muestra en el siguiente esquema que servirá de guía en el desarrollo teórico que

se realizará a lo largo del epígrafe:

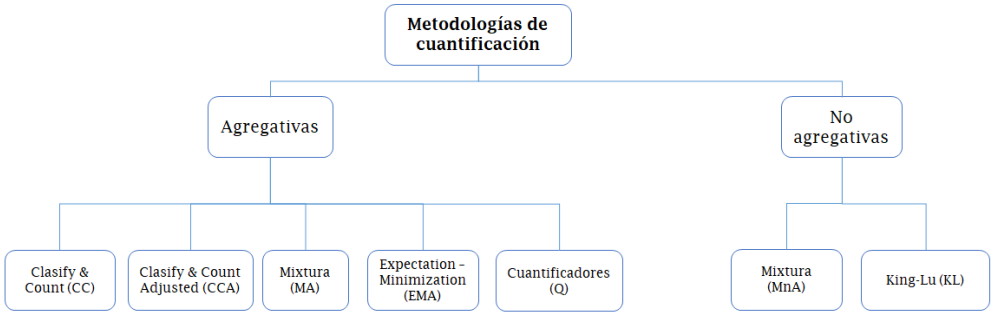

Figura 6: Clasificación técnicas de Cuantificación.

2.1.4.1. Agregativas

1. **Classify and Count (CC)**: está técnica, bautizada con esta denominación por Forman, [For05] en 2005, representa el método más simple e inmediato de *Cuantificación*; conlleva la aplicación directa de un clasificador h^C, entrenado específicamente para la tarea de *Clasificación* asociada, sobre el conjunto de test. La obtención de las probabilidades de cada clase se realiza mediante la agregación de la salida proporcionada por aquel. Existen diversas alternativas de agregación dependiendo de la naturaleza de la salida del clasificador, tal como se esquematiza a continuación:

Tipo de clasificador	Agregador
CRISP	$\mathrm{P}(\hat{\mathrm{i}}) = \frac{1}{\mathrm{M}}\sum_{\mathrm{j}=1}^{\mathrm{M}} I(h^c(x_j) = \mathrm{c_i})$
SOFT	$P(\hat{\imath}) = \frac{1}{M}\sum_{j=1}^{M}[h^c(x_j)]_i$
MIXTO	$P(\hat{\imath}) = \frac{1}{M}\sum_{j=1}^{M} I(\widehat{h^c}(x_j) = c_i)$

Figura 7: Alternativas de agregación en CC.

- **Crisp**: bajo esta denominación se ubican aquellos clasificadores que devuelven la categoría a la que se asigna la observación (v.g.: SVM, árboles de decisión...). La *Cuantificación* se lleva a cabo contando los casos predichos en cada una de las clases.

- **Soft**: se trata de clasificadores que devuelven la probabilidad de pertenencia a cada una de las categorías (v.g.: regresión logística...). La *Cuantificación* se realiza a través del promedio de las probabilidades predichas para cada una de las clases.
- **Mixto**: engloba a todos los clasificadores soft que son modificados mediante algún criterio decisional que transforma la probabilidad[9], emitida en una asignación categórica. Esta transformación habitualmente consiste en la determinación de una colección de puntos, $\{U_1, ..., U_{n-1}\}$ a partir de los cuales se determinan las regiones que delimitan cada clase:

$$U_{i-1} < h^C(x) \leq U_i \Rightarrow \hat{h^C}(x) = c_i$$

Forman en [For06] presenta una variedad de este tipo de métodos que si bien no son aplicados a este método pudieran ser de aplicación:

Método	Descripción
Bayes	El elemento es asignado a la clase positiva si la probabilidad o puntuación es superior a 0.5.
X	Se toma como punto de corte para clasificar las instancias aquel en el que se cortan las curvas TPR y 1-FPR.
Máximo	Se toma como punto de corte aquel que maximiza la diferencia TPR-FPR.
50	El punto en el cual la curva TPR alcanza 0.5.

Figura 8: Diferentes fijaciones de umbrales.

Las debilidades de este enfoque, más alla de que no sea propiamente un método de *Cuantificación*, han sido apuntadas y descritas cuando realizabamos el análisis comparativo entre ambas tareas (*Shift Distribution* y funciones de error diferentes). A la vista de estas debilidades es de esperar que los resultados de este método sean peores que otros métodos más específicos de la *Cuantificación*; los estudios de Forman en [For05], [For06], Gonzales-Castro et al.en , Gao y Sebastiani en han ratificado lo esperado. Es por ello que esta técnica sea utilizada más como benchmark que como una solución efectiva a la tarea de *Cuantificación* tal como se recoge en [For05], [For06] o [Bel+10].

No obstante cabe destacar una voz discrepante en este punto: A. Moreo y F. Sebastiani en [MS20] achacan parte de los malos resultados no tanto a las debilidades apuntadas sino a la falta de adaptabilidad del clasificador aplicado, proponiendo que el clasificador sea entrenado fijando una colección de hiperparámetros optimizados para este tipo de problemas.

[9] En caso de que no estuviesemos ante un clasificador que arrojase probabilidades bastaría transformar los scores por algun tipo de función que lo constriña al intervalo unidad (por ejemplo:logística)

2. **Classify and Count Adjusted (CCA)**: la primera alternativa al enfoque CC busca mejorarlo mediante la aplicación de un ajuste sobre la agregación efectuada por el método CC corrigiendo una de las debilidades expuestas: la variación de la distribución (*Shift Distribution*). El ajuste se fundamenta en la aplicación de la ley de la probabilidad total. La aplicación de ésta genera una colección de igualdades que relacionan linealmente las probabilidades reales de las clases con las probabilidades predichas por el clasificador a través de las desviaciones, medidas en términos de probabilidad, entre lo predicho y lo real:

$$P_{\mathcal{T}}(\hat{c_j}) = \sum_{i=1}^{m} P_{\mathcal{T}}(\hat{c_j}|c_i)P_{\mathcal{T}}(c_i) \quad j \in \{1...m\}$$

donde $P_{\mathcal{T}}(\hat{c_j})$ es el output obtenido en la aplicación directa del clasificador h^C en la muestra de test (es decir, la aplicación del método CC), $P_{\mathcal{T}}(\hat{c_j}|c_i)$ son las probabilidades que recogen las desviaciones en la clasificación en la muestra de test y por último, $P_{\mathcal{T}}(c_i)$ contiene las verdaderas probabilidades de cada una de las clases.

Al sistema anterior hay que añadirle la condiciones de ligadura de un sistema de probabilidades:

$$\sum_{i=1}^{m} P_{\mathcal{T}}(\hat{c_i}|c_j) = 1, \quad 1 \leq j \leq m$$
$$\sum_{i=1}^{m} P_{\mathcal{T}}(c_i) = 1$$

Podemos expresar la anterior colección de igualdades en forma matricial[10]:

[10]Se ha omitido el subíndice de la muestra para no recargar la notación.

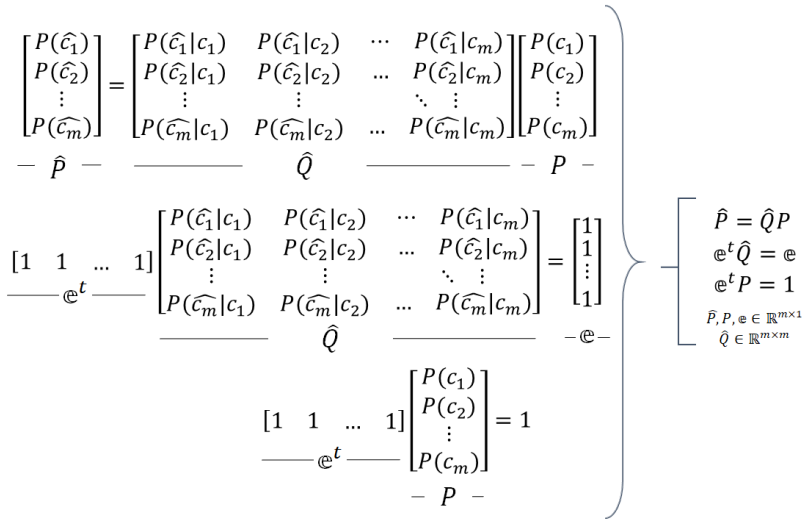

Figura 9: Expresión matricial del método CCA.

El sistema planteado está conformado por $2m + 1$ ecuaciones, si bien independientes son sólo $2m$[11], y $m^2 + m$ incógnitas por lo que el sistema estaría infradeterminado en todos los casos de interés ($m \geq 2$) al disponer de más incognitas que ecuaciones y por tanto, no tendría una única solución existiendo $m^2 - m$ grados de libertad en el sistema.

Para soslayar este problema, y determinar una única solución, es necesario establecer alguna asunción que reduzca el número de incógnitas, en concreto se establece la siguiente:

$$P_{\mathcal{T}}(\hat{c}_j|c_i) = P_{\mathcal{S}}(\hat{c}_j|c_i) \quad \forall\, i, j \in \{1...m\} \tag{1}$$

Es decir, transformamos en observables las probabilidades que recogen la desviación en la asignación de las clases del clasificador en la muestra $\mathcal{T}$ al equipararlas a las observadas en el conjunto de entrenamiento. Se está asumiendo que estas probabilidades condicionadas son estables entre el conjunto de entrenamiento y los diferentes conjuntos de test (no se produce *Shift distribution*) y bajo esta hipótesis se alcanzaría la solución de la tarea; sin embargo hay varias críticas que se deben poner de manifiesto:

- No existe justificación tanto desde un punto de vista teórico como práctico de que esta asunción sea cierta.
- En situaciones en las que alguna de las categorías está poco o nada representada en la muestra de entrenamiento este método no funcionaría adecudamante. Este hecho ya lo pone de manifiesto Forman en

[11] La última ecuación del sistema $\hat{P} = \hat{Q}P$ se reconstruye mediante la adición de las m-1 anteriores y la incorporación de las restricciones.

[For06] introduciendo diferentes posibles soluciones que analizaremos con posterioridad en la particularización al caso binario.

- Al igual que en cualquier método de estimación las magnitudes están sometidas a un error de estimación que evitará obtener resultados perfectos.

A pesar de las debilidades apuntadas el método CCA ofrece mejores resultados que el CC en toda la literatura consultada al respecto. Desde un punto de vista práctico la determinación de las probabilidades, $P_{\mathcal{S}}(\hat{c_j}|c_i)$, se realiza de acuerdo a dos posibilidades, a saber:

- Reserva de Muestra o *Hold-Out*: consiste en separar un conjunto de observaciones del conjunto de entrenamiento para poder llevar a cabo la estimación de las probabilidades condicionadas.
- Validación cruzada o *Cross-validation*: en situaciones en las que el anterior método no es posible dado que el conjunto de entrenamiento es pequeño es preferible aplicar la validación cruzada, construyendo las probabilidades mediante algún estadístico central, preferiblemente la media, de las magnitudes obtendidas en cada una de las particiones (*folds*). Este es el enfoque más usado en la literatura, así Forman en [For05], [For06] y [For07] estima las probabilidades para un problema binario utilizando una validación cruzada con 50 *folds*, Bella et al.en [Bel+10] también se decantan por la misma configuración, mientras que Barranquero et al.en [Bar+12], Pérez-Gallego en [Pér+18] o Xue et al. en [XW09] se decantan por utilizar 10 *folds*.

Dado que nos centraremos en el problema binario, $c \in \{1, -1\}$, desarrollaremos a continuación la aplicación de esta metodología en este tipo de problema:

$$\begin{cases} \begin{bmatrix} P_{\mathcal{T}}(\hat{1}) \\ P_{\mathcal{T}}(\hat{-1}) \end{bmatrix} = \begin{bmatrix} P_{\mathcal{T}}(\hat{1}|1) & P_{\mathcal{T}}(\hat{1}|-1) \\ P_{\mathcal{T}}(\hat{-1}|1) & P_{\mathcal{T}}(\hat{-1}|-1) \end{bmatrix} \begin{bmatrix} P_{\mathcal{T}}(1) \\ P_{\mathcal{T}}(-1) \end{bmatrix} \\ \\ P_{\mathcal{T}}(1) + P_{\mathcal{T}}(-1) = 1 \\ P_{\mathcal{T}}(\hat{1}|1) + P_{\mathcal{T}}(\hat{-1}|1) = 1 \\ P_{\mathcal{T}}(\hat{1}|-1) + P_{\mathcal{T}}(\hat{-1}|-1) = 1 \end{cases}$$

Simplificando y aplicando la asunción (1):

$$P_{\mathcal{T}}(\hat{1}) = P_{\mathcal{S}}(\hat{1}|1)P_{\mathcal{T}}(1)+P_{\mathcal{S}}(\hat{1}|-1)[P_{\mathcal{T}}(-1)] = P_{\mathcal{S}}(\hat{1}|1)P_{\mathcal{T}}(1)+P_{\mathcal{S}}(\hat{1}|-1)[1-P_{\mathcal{T}}(1)]$$

Por lo que la probabilidad $P_{\mathcal{T}}(1)$ se obtiene como:

$$P_{\mathcal{T}}(1) = \frac{P_{\mathcal{T}}(\hat{1}) - P_{\mathcal{S}}(\hat{1}|-1)}{P_{\mathcal{S}}(\hat{1}|1) - P_{\mathcal{S}}(\hat{1}|-1)} = \frac{P_{\mathcal{T}}(\hat{1}) - FPR}{TPR - FPR}$$

donde en la última igualdad hemos utilizado la notación en términos de errores característica de los problemas de clasificación binaria[12]. Al igual que ocurría con el enfoque CC la naturaleza de la salida del clasificador tiene su influencia en la aplicación del método:

a) **Crisp**: en el trabajo fundacional de la *Cuantificación* [For05] Forman aplica el enfoque CCA sobre un clasificador SVM en un problema de *Cuantificación* binaria:

$$P_{\mathcal{T}}(1) = \frac{P_{\mathcal{T}}(\hat{1}) - FPR}{TPR - FPR}$$

b) **Soft**: en el caso que el clasificador arroje probabilidades de pertenencia a las clases Bella et al. en [Bel+10] adaptan el método anterior computando las magnitudes FPR y TPR de acuerdo a las siguientes expresiones:

$$TPR = \frac{1}{M_1} \sum_{i|c_i=1} (h^C(x_i))$$

$$FPR = \frac{1}{M_{-1}} \sum_{i|c_i=-1} (h^C(x_i))$$

donde hemos denotado por M_1 y M_{-1} al número de elementos del conjunto de entrenamiento que están en la clase 1 y -1 respectivamente.

c) **Mixtos**: aunque se introdujeron con antelación en la presentación del método CC es aquí, en la aplicación del método CCA, donde han sido utilizados en la práctica. Forman los introduce como una solución con la que soslayar una de las problematicas que se presenta al determinar los parámetros TPR y FPR: la descompensación de la clase positiva frente a la negativa.

Las diferentes alternativas ya fueron mencionadas en el punto anterior dedicado al método CC, por lo que aquí nos centraremos en exponer un criterio nuevo y exclusivo de la metodología CCA que Forman introdujo en [For07] y que denomina *Median Sweep*.

Bajo esta metodología para cada uno de los posibles umbrales U_i se determina la probabilidad que resultaría de aplicar el método CCA:

$$P_{\mathcal{T}}(1)_i = \frac{P_{\mathcal{T}}(\hat{1}) - FPR(U_i)}{TPR(U_i) - FPR(U_i)}$$

La probabilidad final se obtiene tomando la mediana de las anteriores probabilidades:

$$P_{\mathcal{T}}(1) = Med\{P_{\mathcal{T}}(1)_i\}$$

[12]FPR: False Positive Rate, TPR: True Positive Rate.

De todos los métodos presentados por Forman es éste el que mejor resultados ofrece en términos de precisión dentro del diseño experimental planteado por el autor.

3. **Mixtura agregativa (MA).** Recibe su nombre de la técnica homónima utilizada en teoría de probabilidad para obtener una nueva distribución, F, a partir de la combinación convexa de otras ya existentes, F_i; es decir, se pretende generar una nueva distribución siguiendo el siguiente esquema:

$$F(x) = \sum_{i=1}^{m} F_i(x)\omega_i$$

En su aplicación a la *Cuantificación* el papel de F, lo juega la distribución de la puntuación del clasificador[13] h^C en la muestra de test $\mathcal{T}$, la denotaremos por $F_{\mathcal{T}}$. Esta distribución incorpora las distribuciones de puntuación de cada categoría en función de sus prevalencias o probabilidades de interés, matemáticamente lo expresaríamos como sigue:

$$F_{\mathcal{T}}(s) = P_{\mathcal{T}}(h^C \leq s) = \sum_{i=1}^{m} P_{\mathcal{T}}((h^C \leq s) \cap c_i) = \sum_{i=1}^{m} P_{\mathcal{T}}((h^C \leq s)|c_i)P_{\mathcal{T}}(i) =$$
$$= \sum_{i=1}^{m} F^i_{\mathcal{T}}(s)P_{\mathcal{T}}(i)$$

Donde hemos denotado por $F^i_{\mathcal{T}}(s)$ la distribución de la puntuación del clasificador para la clase c_i en la muestra de test y $P_{\mathcal{T}}(i)$ las probabilidades de cada una de las clases que se corresponden con los pesos de la mixtura. Bajo este planteamiento el objetivo de MA será la determinación de estos pesos de tal forma que generen la mejor aproximación a la distribución $F_{\mathcal{T}}$.
Sin embargo, para llevar a cabo este objetivo se debe realizar alguna asunción adicional con el fin de reducir el número de elementos indeterminados. La asunción establecida asume que las distribuciones de los clasificadores en cada clase se mantienen constantes respecto al conjunto de entrenamiento:

$$F^i_{\mathcal{T}}(s) = F^i_{\mathcal{S}}(s) \quad 1 \leq i \leq m$$

Con esta asunción podemos plantear un programa de optimización en el

[13] Tácitamente estamos imponiendo que el clasificador sea de tipo Soft.

que se materializa el método MA:

$$\begin{aligned} &\min_{P_{\mathcal{T}}(1)\ldots P_{\mathcal{T}}(m)} d(F_{\mathcal{T}}, \sum_{i=1}^{m} F_{\mathcal{S}}^{i} P_{\mathcal{T}}(i)) \\ &\text{Sujeto a :} \\ &P_{\mathcal{T}}(i) \geq 0,\ 1 \leq i \leq m \\ &\sum_{i=1}^{m} P_{\mathcal{T}}(i) = 1, \end{aligned} \tag{2}$$

donde d es la medida elegida para cuantificar la diferencia entre la distribución en la muestra de test y la mixtura. Este mismo planteamiento teórico podría rescribirse con la función de densidad asociada sustituyendo en (2) $F_{\mathcal{T}}$ y $F_{\mathcal{S}}^{i}$ por sus derivadas.

La concreción del planteamiento deja, no obstante, ciertas elecciones al investigador que es conveniente explicitar:

- **Función de distribución vs. función de densidad**. Tal cómo mencionabamos, el método MA puede aplicarse tanto a la función de distribución como a la función de densidad de las puntuaciones del clasificador; así, Gonzalez-Castro en [GAA13] y Maletzke et al. [Mal+19] prefieren usar las funciones de densidad en los experimentos realizados, mientras que Forman en [For05] y [For06] se decanta por el uso de la función de distribución.

  La elección de uno u otro enfoque depende de las preferencias del investigador, si bien habría que tener en cuenta que en el caso de utilizar las funciones de densidad será necesario introducir un nuevo parámetro que regule la discretización: el número de *bins*. La complejidad del proceso se incrementa y será necesario analizar la influencia de este parámetro sobre los resultados. La elección de un número de *bins* supondrá establecer un dilema precisión-estabilidad; así, un número muy elevado de *bins* redundaría en una mejor precisión a costa de incorporar grupos con poca masa crítica que incorporarían inestabilidad en la la estimación del histograma, mientras que la reducción del número de *bins* reduciría la variabilidad de las estimaciones penalizando la precisión de la estimación. Con el fin de paliar esta elección unitaria Gonzalez-Castro en [GAA13] realizan diversos experimentos variando el número de bins: desde 10 hasta 110 con incrementos de tamaño 10 hasta obtener 11 aproximaciones diferentes[14].

  A pesar de lo apuntado con anterioridad, es mayoritaria la utilización del enfoque de función de densidad, e intuimos que es debido a la

[14] Finalmente, y tomando una decisión salomónica, obtienen la predicción de la probabilidad como la mediana de las 11 predicciones realizadas.

amplia gama de medidas, d, que aporta esta aproximación frente a la utilización de la función de distribución.

- **Estimación de $F^i_{\mathcal{S}}$**. La estimación de las distribuciones de cada clase se realizan sobre el conjunto de entrenamiento mediante validación cruzada. Forman en [For05] y [For06] usa 50 *folds* de validación cruzada para evitar el sobreajuste de las distribuciones obtenidas.
- **Distancia elegida**. Son diversas las alternativas que se han tomado al respecto y en Maletzke et al. [Mal+19] se hace un análisis completo sobre el rendimiento de gran parte de ellas. En la tabla inferior se muestra un compendio de las principales alternativas tomadas en la literatura revisada:

Distancia	Expresión	Comentario	Artículo
Hellinger	$2\sqrt{1-\sum_{i=1}^{b}\sqrt{F_i G_i}}$	Se utiliza para dos histogramas F y G con el mismo número de bins.	i. **Class distribution estimation based on the hellinger distance.** ii. **DyS: A Framework for Mixture Models in Quantification.**
PP-Area	Area que queda bajo las curvas P-P de las dos distribuciones de probabilidad .	Se utiliza para distribuciones de probabilidad y no para histogramas. La minimización de esta métrica es equivalente a la minimización de la distancia de Manhattan.	i. **Quantifying Counts, Costs, and Trends Accurately via Machine Learning**
Kolmogorov-Smirnoff	$Max_x\lvert F(x)-G(x)\rvert$	Se utiliza para distribuciones de probabilidad y no para histogramas.	
Manhattan	$\sum_{i=1}^{b}\lvert F_i-G_i\rvert$	Se utiliza para dos histogramas F y G con el mismo número de bins.	**i. Quantifying Counts, Costs, and Trends Accurately via Machine Learning** **ii. DyS: A Framework for Mixture Models in Quantification.**
Coseno	$\frac{\sum_{i=1}^{b}F_i G_i}{\sqrt{\sum_{i=1}^{b}F_i^2}\sqrt{\sum_{i=1}^{b}G_i^2}}$	Se utiliza para dos histogramas F y G con el mismo número de bins.	**i. DyS: A Framework for Mixture Models in Quantification.**
Jaccard	$\frac{\sum_{i=1}^{b}(F_i-G_i)^2}{\sum_{i=1}^{b}F_i^2+\sum_{i=1}^{b}G_i^2-\sum_{i=1}^{b}F_i G_i}$	Se utiliza para dos histogramas F y G con el mismo número de bins.	**i. DyS: A Framework for Mixture Models in Quantification.**
Cuadrática	$\sum_{i=1}^{b}(F_i-G_i)^2$	Se utiliza para dos histogramas F y G con el mismo número de bins.	**i. DyS: A Framework for Mixture Models in Quantification.**

Figura 10: Distancias elegidas para la Mixtura Agregativa.

- **Método de optimización**. Dado que en los artículos consultados este método sólo ha sido puesto en práctica para problemas binarios el algoritmo de busqueda elegido parte de la busqueda explícita sobre una partición del intervalo unidad.

La comparativa de los resultados de esta metodología frente a las anteriores analizadas arroja dos conclusiones:

- Es mejor, en términos de error, al método CC en todos los documentos analizados.
- Esta relación no es tan obvia cuando lo enfrentamos al CCA, mientras que en [GAA13] el método MA supera al CCA en todos los conjuntos de entrenamiento no ocurre lo mismo en el caso de los análisis de

Forman en [For06] donde el método es batido por el CCA con determinadas elecciones del punto de corte, por ejemplo *Median Sweep*. No obstante estos resultados de Forman no son uniformes sino que depende del nivel de prevalencia. A la vista de esto, y de no haber estudios definitivos que hagan comparaciones homogéneas no podemos inclinarnos en favor de una u otra metodología.

4. **Expectation-Minimization (EMA).** Desde un punto de vista teórico el algoritmo *Expectation – Maximization* (EM) es un método iterativo que permite obtener estimadores máximo verosímiles, al menos de carácter local, en modelos probabilísticos. Su utilidad está condicionada a la situación en la que existen variables no observables, también denominadas latentes, que forman parte del proceso generador de datos.
   El método consta de las dos etapas que dan nombre al proceso, a saber[15]:

   - *Expectation (E)*: en esta etapa se construye la función que será objeto de maximización en el paso posterior (f_t). Esta función se construye mediante la esperanza del logaritmo de la verosimilitud.
   - *Maximization (M)*: la construcción de la función anterior es posible gracias a la maximización de la función de la etapa anterior (f_{t-1}). Esta optimización arroja los parámetros con los que definir dicha función.

   El trabajo de Saerens et al. [SLD02] adapta este algoritmo al contexto de la *Cuantificación*. Su derivación parte de la formulación habitual de la verosimilitud para el conjunto de test, que es allí donde se requiere conocer la distribución de las clases:

$$L(\mathcal{T}|\pi) = \prod_{k=1}^{M}\prod_{i=1}^{m} P_{\mathcal{T}}(x_k, c_i)^{y_{ki}} = \prod_{k=1}^{M}\prod_{i=1}^{m}(P_{\mathcal{T}}(x_k|c_i)P_{\mathcal{T}}(c_i))^{y_{ki}}$$

   donde por X hemos denotado el vector de rasgos de la muestra de test, Y es una variable indicador de cada una de las clases[16] en la muestra de test y π es el vector de probabilidades en el que estamos interesados $\pi = (P_{\mathcal{T}}(1), ..., P_{\mathcal{T}}(m))$.
   La aplicación de logaritmos conduce a la expresión habitual sobre la que se aplica el algoritmo EM:

$$l(X, Y|\pi) = \sum_{k=1}^{M}\sum_{i=1}^{m} y_{ki} log(P_{\mathcal{T}}(c_i)) + \sum_{k=1}^{M}\sum_{i=1}^{m} y_{ki} log(P_{\mathcal{T}}(x_k|c_i))$$

   Puesto que las etiquetas, y_{ki}, no son observables en la muestra de test, para llevar a cabo el paso *Expectation* se reemplaza la suma anterior por

[15] De cara a su mejor intepretación ha de suponerse que estamos en una iteración intermedia.
[16] Es decir toma el valor 1 si la observación k-ésima tiene la clase i-ésima, en el caso contrario toma el valor 0.

su esperanza respecto a $P(Y|X,\pi)$:

$$l(X,Y|\pi) \rightarrow \mathbb{E}[l(X,Y|\pi)|X,\pi]$$

por lo que obtenemos:

$$\mathbb{E}[l(X,Y|\pi)|X,\pi] = \sum_{k=1}^{M}\sum_{i=1}^{m}\mathbb{E}[y_{ki}|x_k,\pi]log(P_{\mathcal{T}}(c_i))+\sum_{k=1}^{M}\sum_{i=1}^{m}\mathbb{E}[y_{ki}|x_k,\pi]log(P_{\mathcal{T}}(x_k|c_i))$$

La obtención de las esperanzas condicionadas se realiza de la siguiente forma:

$$\mathbb{E}[y_{ki}|x_k,\pi] = P_{\mathcal{T}}(y_{ki}=1|x_k,\pi)$$

Dado que π es desconocida la sustituiremos por el valor obtenido en el paso anterior del algoritmo, el cual denotaremos por π^t, de esta manera:

$$\mathbb{E}[y_{ki}|x_k,\pi] = P_{\mathcal{T}}(y_{ki}=1|x_k,\pi^t) = P^t_{\mathcal{T}}(c_i|x_k) \tag{3}$$

Adicionalmente impondremos una asunción adicional:

$$P_{\mathcal{T}}(X|c_i) = P_{\mathcal{S}}(X|c_i)$$

La cual nos permitirá rescribir (3) como sigue:

$$P^t_{\mathcal{T}}(c_i|x_l) = \frac{\frac{P^t_{\mathcal{T}}(c_i)}{P_{\mathcal{S}}(c_i)}P_{\mathcal{S}}(c_i|x_l)}{\sum_{j=1}^{m}\frac{P^t(c_j)}{P_{\mathcal{S}}(c_j)}P_{\mathcal{S}}(c_j|x_l)} \quad l=1...M,\ i=1...m$$

El paso M permitirá obtener la probabilidad a posteriori de cada clase como promedio de las probabilidades anteriores sobre todas las observaciones del conjunto de test, es decir:

$$P_{\mathcal{T}}{}^t(c_i) = \frac{1}{M}\sum_{l=1}^{M}P^{t-1}_{\mathcal{T}}(c_i|x_l)$$

En resumen, el algoritmo tal como se explicita en [SLD02] adopta la siguiente forma:

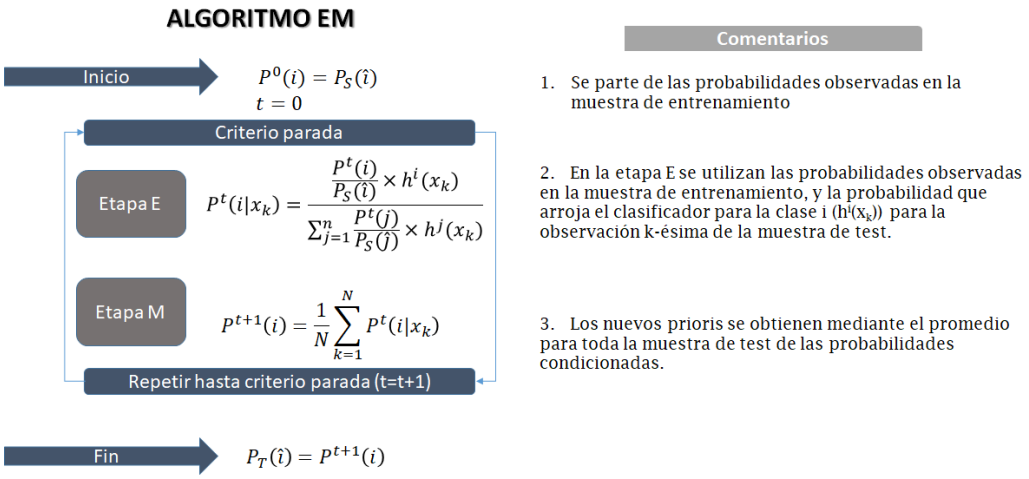

Figura 11: Algoritmo Expectation-Maximization.

El proceso se realizará tantas veces como se considere necesario hasta llegar a la convergencia si bien como mencionan los autores ésta puede ser a un óptimo local y no global. En el análisis de comparativa que los autores muestran se evidencia que mejora el método benchmark que eligen, el CCA.

5. **Cuantificadores (Q).** Se trata de la técnica que más actividad investigadora está concentrando actualmente en el ámbito de la *Cuantificación*; está enfocada al desarrollo de algoritmos especializados a la *Cuantificación* tomando como base un clasificador y ajustando su función de pérdida en función de alguna medida de error propia de la *Cuantificación*.

   Se han generado diversas alternativas, a continuación presentamos las más relevantes:

   - **SVM**. Diversos trabajos han abordado las adaptaciones de las máquinas de vector soporte, SVM, a los problemas de *Cuantificación*. Tanto Sebastiani et al. en [GS15] y en [Seb18c], como Barranquero et al. en [BDC15] hacen uso de una adaptación del clásico SVM llevada a cabo por Thorsten Joachims que permite ser aplicado a funciones no-lineales de carácter multivariante y que puedan ser calculadas desde una tabla de contingencia. Las principales diferencias entre el SVM clásico y el desarrollado por T. Joachims, tanto en la estimación como en la aplicación, se esquematizan a continuación.

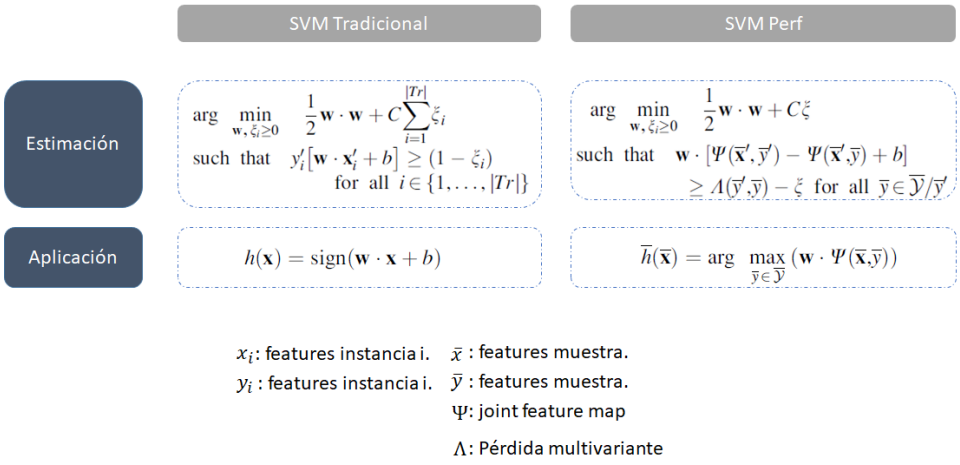

Figura 12: Comparativa SVM clásico & SVM Perf.

Como se puede observar:

a) La función de pérdida Λ aparece explícitamente en la especificación del problema en el caso del SVM-perf.

b) La evaluación y estimación en el caso de SVM perf es a nivel muestra en contraposición con el SVM clásico donde se procesa cada instancia independientemente.

Existen diferencias en la aplicación llevada a cabo por Sebastiani y la que desarrolla Barranquero en lo que supone la elección de la función de pérdida. Mientras Sebastiani et al. en [GS15] y en [Seb18c] utilizan la distancia de Kullback-Leibler (KLD) lo que le permite focalizar la optimización desde el punto de vista de la *Cuantificación*, Barranquero incorpora como medida de error, una inspirada en los F-Score que auna tanto el objetivo de *Clasificación* como el de *Cuantificación* Q_β:

$$Q_\beta = (1+\beta)^2 \frac{cperf \times qperf}{\beta^2 cperf + qperf}$$

donde *cperf* y *qperf* son medidas de desempeño de *Cuantificación* y *Clasificación*. En este documento los autores utilizan como medidas de desempeño de *Cuantificación* el Error Absoluto (AE) y el Error Cuadratico Relativo (ES)[17].

- **Redes Neuronales**. Esuli et al en [EFS18] desarrollan una algoritmo, denominado QuaNet, que combina dos redes neuronales de diverso tipo para construir un cuantificador. La arquitectura desarrollada consta de dos redes neuronales: una inicial que hace el papel

[17] En el epígrafe 2.1.6 se hace una breve descripción de las principales medidas de desempeño en el ámbito de Cuantificación

del clasificador y otra segunda que recibe la información de la primera y lleva a cabo el proceso de *Cuantificación*. De forma similar en [Qi+21] se expone un marco general para la aplicación del deep learning en el ámbito de la *Cuantificación*.

- **Árboles de decisión**. El primer trabajo sobre la adaptación de los árboles de decisión se lleva a cabo en Milli et al. [Mil+13] en el cual se adapta el algoritmo de aprendizaje habitual de los árboles para que en la función de pérdida se incluya también una medida de la bondad de la *Cuantificación*. Esto requerirá adaptar los dos procesos clave en la estimación de un árbol de decisión: la selección del punto de corte óptimo y el criterio de parada.

  a) Selección de punto de corte óptimo. Para la selección del punto de corte los autores valoran dos posibilidades, por un lado una medida que sólo tenga en cuenta en la función de pérdida el objetivo de clasificación y otra novedosa en la que se utiliza una medida que combina tanto el desempeño de la propia clasificación como de la *Cuantificación* que es la que explicitamos a continuación:

  $$E = |FP - FN|^2 = |FP - FN| \times |FP + FN|$$

  Tal como se ha comentado anteriormente en el documento, un buen clasificador tendrá un bajo valor para el término $|FP+FN|$ mientras que un buen cuantificador mantendrá un reducido valor para el término $|FP - FN|$. La selección de punto de corte y la consiguiente escisión del nodo se realiza a través de la ganancia entre los nodos:

  $$\Delta = E_{padre} - E_{hijo}$$

  b) Criterio de parada. El establecimiento del criterio de parada se establece en aquel caso en el que la ganancia, Δ, no es positiva.

- **K-Nearest Neighborhood (NN)**. Barranquero et al. en [Bar+12] plantean la adaptación de metodologías NN para la *Cuantificación*. Ensayan tanto un NN simple como uno ponderado.

2.1.4.2. No Agregativas Las técnicas no agregativas son aquellas en las que no se utiliza un clasificador. Para solventar este hecho se apoyan en las distribuciones de los rasgos en la muestra de entrenamiento, es decir, mantienen un espíritu similar al visto en las técnicas MA o CCA sustituyendo las distribuciones de los clasificadores por las distribuciones de los rasgos.

La aplicación de estas técnicas tiene la ventaja obvia de no tener que desarrollar un clasificador, si bien el precio que hay que pagar es la elevación de la complejidad al apoyarse en el espacio de rasgos $\mathcal{X}$ cuya dimensionalidad es elevada y ello conlleva un incremento de la complejidad computacional.

A pesar de la desventaja apuntada es una técnica que pudiera ser muy útil, sin embargo, no abunda demasiada literatura al respecto. Hasta el momento existen dos tipos de algoritmos para este tipo de enfoques:

- **Mixture Agregation (MnA)**. *Mutatis mutandi* se trata del mismo método al MA pero teniendo únicamente en cuenta las distribuciones de las features. Esta desarrollado en Gonzalez-Castro et al. [GAA13] y es denominado por los autores como método HDx dado que como función de distancia utilizan la de Hellinger aplicada a las distribuciones de rasgos (de ahí la x). Al igual que en el caso agregativo el objetivo es determinar la probabilidad que haga mínima la distancia de la distribución de los rasgos en el conjunto de test y la mixtura de distribuciones en el conjunto de entrenamiento, si bien la principal novedad es que la distribución de los rasgos es multidimensional:

$$\min_{P_{\mathcal{T}}(1)\ldots P_{\mathcal{T}}(n)} HD(\mathbf{F}_{\mathcal{T}}, \sum_{i=1}^{n} \mathbf{F}_{\mathcal{S}}^{i} P_{\mathcal{T}}(i))$$

$$\text{Sujeto a :}$$

$$P_{\mathcal{T}}(i) \geq 0, \; \forall\, i$$

$$\sum_{i=1}^{n} P_{\mathcal{T}}(i) = 1,$$

  Donde $\mathbf{F}_{\mathcal{T}}$, $\mathbf{F}_{\mathcal{S}}^{i}$ hacen referencia a las distribuciones multidimensional de los rasgos en la muestra de test y entrenamiento respectivamente. Uno de los problemas derivados de este enfoque es la alta dimensionalidad del problema, lo que por un lado empobrece la caracterización de la distribución, ya que muchos de los bins estarán vacíos o bien contaran con pocas observaciones, lo que implicará unas estimaciones volátiles.

- **Método King & Lu**. Este método parte de un planteamiento análogo al que se hacía en el caso de los CCA, es decir, partiendo de la ley de la probabilidad total ligar las probabilidades de las clases con las de algún elemento observable. Los autores King & Lu [KL08] o en el más reciente de Hopkins and King [HK10] utilizan los rasgos para seguir un esquema similar, así para un determinado rasgo, x, se obtiene la siguiente relación:

$$P_{\mathcal{T}}(x) = \sum_{i=1}^{m} P_{\mathcal{T}}(x|c_i) P_{\mathcal{T}}(c_i)$$

  Dado que no se dispone de la información de la distribución de las probabilidades condicionadas en la muestra de test, se realiza la siguiente asunción:

$$P_{\mathcal{T}}(x|c_i) = P_{\mathcal{S}}(x|c_i)$$

Una vez incorporada nos encontramos ante un sistema lineal que puede ser resuelto numéricamente.
Los principales problemas a los que se enfrenta esta metodología son muy similares a los hasta ahora vistos:

- El tamaño del sistema lineal puede ser elevado dependiendo del número de rasgos y la dimensionalidad de éstos. Este hecho conlleva un incremento en la complejidad del algoritmo de resolución.
- La matriz asociada está muy poco informada, manteniendo gran parte de la misma a cero (es lo que se denomina *sparse matrix*) lo cual requerirá la aplicación de algoritmos específicos para optimizar la resolución de los mismos. Adicionalmente, y relacionado con este punto, pudieran surgir problemas numéricos en la resolución.

2.1.5. Diseños experimentales

Los diseños experimentales también muestran diferencias significativas con los que habitualmente se realizan en tareas de clasificación. Así, mientras que en un problema de *Clasificación* sólo es necesario disponer de un conjunto de test, en la *Cuantificación* no es razonable disponer de un único conjunto sobre el que evaluar la metodología, ya que daría lugar a una única valoración. Es necesario disponer de un colección de conjuntos de test sobre los que medir la bondad de la metodología de *Cuantificación* diseñada.

Actualmente existen diversos repositorios de información en los que se pueden obtener colecciones de conjuntos sobre los que hacer mediciones del desempeño de los cuantificadores[18], aunque todos ellos son conjuntos que provienen de problemas de *Clasificación* no propiamente de la *Cuantificación*. En la mayor parte de los trabajos consultados la opción elegida a la hora de llevar a cabo la experimentación es utilizar las fuentes gratuitas, aunque existen casos como los de Forman [For05], [For06] en los que se decantan por conjuntos artificiales[19]. La utilización de este tipo de conjuntos permite generar, de forma natural, diferentes niveles de probabilidad sobre los que comprobar la idoneidad del cuantificador. Este aspecto a veces es difícil de conseguir directamente si se acude a conjuntos reales y requiere de algún tipo de remuestreo para obtener diferentes niveles de prevalencia de las clases.

El diseño de experimentación canónico es el siguiente:

[18] Por ejemplo, y será el que se utilice en esta memoria los conjuntos de datos depositados en la web Machine Learning Repository https://archive.ics.uci.edu/ml/datasets.php

[19] Generados mediante algún proceso de simulación probabilística

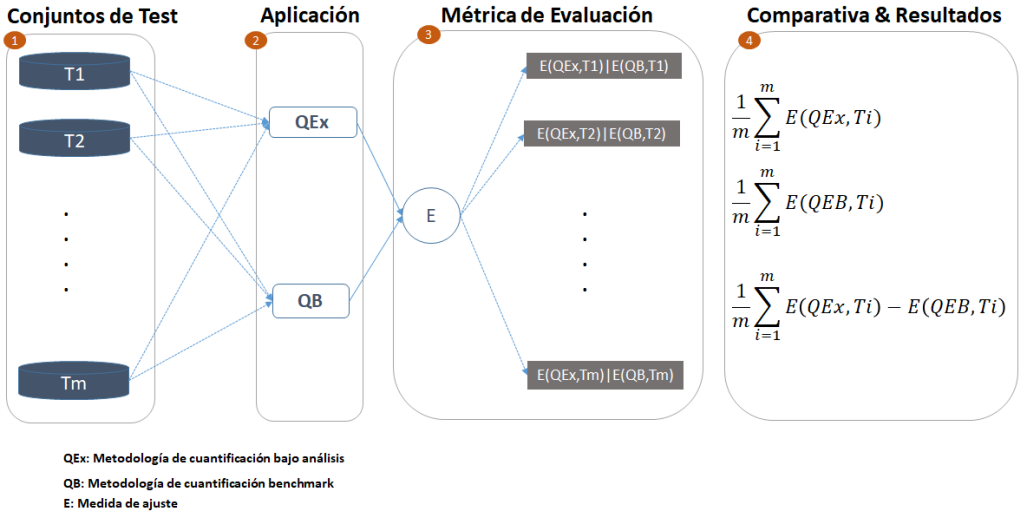

Figura 13: Diseño experimental en Cuantificación.

1. **Conjuntos de test**. Tal como mencionabamos al inicio del epígrafe una de las características de la experimentación en la *Cuantificación* es la necesidad de multiples conjuntos sobre los que testar la metodología. La mayor parte de conjuntos de datos reales que se disponen no son propios de esta tarea sino que provienen de la *Clasificación* en su mayor parte, y por tanto sólo se dispone de un conjunto de test, por lo que generalmente la generación de esta multiplicidad debe realizarse de forma artificial mediante remuestreo.

2. **Aplicación**. Para cada uno de los conjuntos de test disponibles se aplica la metodología de *Cuantificación* bajo estudio, *Qex*. Esta metodología debe ser contrastada con algún tipo de metodología benchmark *QB*; habitualmente se toman como metodologías benchmark las más sencillas de todas las presentadas: CC y CCA.

3. **Métricas de Evaluación**. En esta etapa se fija la medida ajuste con la que se llevará a cabo la evaluación de las metodologías aplicadas en el punto anterior. Dado que en el siguiente punto haremos un breve bosquejo de las principales medidas de evaluación no nos extenderemos más en este punto.

4. **Comparativa** & **Resultados** A partir de los resultados obtenidos en cada una de las muestras tanto para la metodología bajo análisis como para la benchmark se llevan a cabo análisis estadísticos que verifiquen si las diferencias observadas son significativas. Esta etapa no difiere de un análisis de resultados al uso que se puede llevar a cabo en la *Clasificación*.

2.1.6. Métricas de evaluación

Los ensayos empíricos revisados hacen uso de una colección bastante estándar de métricas de evaluación que son analizadas a través de múltiples conjuntos de test. A continuación se presentan las más habituales junto con los principales documentos donde son utilizadas.

Es importante notar que las medidas se presentarán genéricamente para un problema de *Cuantificación* de multicategoría, explicitando la medida para una clase genérica c_i, incorporando adicionalmente la agregación de ésta a todas las categorías.

Medida	Descripción	Expresión	Principales Artículos
E	Error	$\frac{1}{M}\sum_{i=1}^{M} P(i) - P(\hat{i})$	i. Quantifying Counts, Costs, and Trends Accurately via Machine Learning
MAE	Error absoluto	$\frac{1}{M}\sum_{i=1}^{M} \lvert P(i) - P(\hat{i})\rvert$	i. Counting Positives Accurately Despite Inaccurate Classification. Ii. Quantifying Counts, Costs, and Trends Accurately via Machine Learning. Iii.Quantification via Probability Estimators iv.Adjusting the Outputs of a Classifier to New a Priori Probabilities: A Simple Procedure v.DyS: A Framework for Mixture Models in Quantification vi. Class Distribution Estimation based on the Hellinger Distance. vii. A Recurrent Neural Network for Sentiment Quantification. viii. A Framework for Deep Quantification Learning. ix. Quantification oriented learning based on reliable classifiers. x. Tweet Sentiment: From Classification to Quantification
NAE	Error Absoluto Normalizado	$\frac{\sum_{i=1}^{M} \lvert P(i) - P(\hat{i})\rvert}{2(1 - \min(P(i)))}$	i. Tweet Sentiment: From Classification to Quantification
RAE	Error Absoluto Relativo	$\frac{1}{M}\sum_{i=1}^{M} \frac{\lvert P(i) - P(\hat{i})\rvert}{P(i)}$	i. Class Distribution Estimation based on the Hellinger Distance. Ii. A Recurrent Neural Network for Sentiment Quantification. Iii. A Framework for Deep Quantification Learning. Iv. Tweet Sentiment: From Classification to Quantification.
ERS	Error Relativo Simétrico	$\sum_{i=1}^{M} \frac{\lvert P(i) - P(\hat{i})\rvert}{P(i) + P(\hat{i})}$	
NRAE	Error Absoluto Relativo Normalizado	$\frac{\sum_{i=1}^{M} \frac{\lvert P(i) - P(\hat{i})\rvert}{P(i)}}{M - 1 + \frac{1 - \min(P(i))}{\min(P(i))}}$	i. Combining instance selection and self-training to improve data stream quantification.

Figura 14: Inventario de métricas de evaluación Cuantificación I.

Este primer conjunto de medidas representan las más sencillas de aplicación, y por ello suelene ser lás comunmente utilizadas:

Error o Sesgo (E). Se trata de la medida más sencilla, mide la diferencia de la estimación entre las probabilidades estimadas y reales de cada una de las clases. Su rango de definición cubre toda la recta real positiva. Su principal problema reside en la potencial compensación de errores de signo contrario lo que conllevaría una falsa sensación de certeza.

Media Error Absoluto (MAE). Es la medida más popular en los estudios analizados siendo su presencia mayoritaria. Mitiga la anterior debilidad apuntada incorporando el valor absoluto en su computo. Al igual que la anterior su rango de definición yace en la recta real positiva. El principal problema que nos encontramos es que es insensible a las magnitudes de los errores absolutos frente a las probabilidades que se pretenden predecir.[20].

[20] Es decir se valora de igual forma una error absoluto de 0.01 aproximando una probabilidad

Error Relativo absoluto (RAE). Solventa el incoveniente señalado para la medida MAE al tener en cuenta la probabilidad que se pretende predecir. Dado que se está incorporando un cociente cabe la posibilidad de anular el denominador y por tanto provocar una indeterminación por lo que a veces se recurre a un suavizado de las probabilidades:

$$P_s(i) = \frac{\epsilon + P(i)}{\epsilon M + \sum_{i=1}^{M} P(i)}$$

Medida	Descripción	Expresión	Principales Artículos
SE	Error Cuadrático Medio	$\frac{1}{M}\sum_{i=1}^{M}\|P(i) - P(\hat{i})\|^2$	
ME	Raiz del Error Cuadrático Medio	$\sqrt{\frac{1}{M}\sum_{i=1}^{M}\|P(i) - P(\hat{i})\|^2}$	I.Quantification via Probability Estimators. Ii. A Method of Automated Nonparametric Content Analysis for Social Science
Entropia Normalizada		$CE(P,\hat{P}) - CE(P,P)$	i. Counting Positives Accurately Despite Inaccurate Classification. Ii. Quantifying Counts, Costs, and Trends Accurately via Machine Learning
KL	Kullback-Leibler	$\sum_{i=1}^{M} P(i)Ln(\frac{P(i)}{P(\hat{i})})$	i.Quantification Trees ii. Ordinal Text Quantification. Iii. A Recurrent Neural Network for Sentiment Quantification. Iv. A Framework for Deep Quantification Learning v. Tweet Sentiment: From Classification to Quantification. Vi. Online Optimization Methods for the Quantification Problem vii. Semi-Supervised Learning of Class Balance under Class-Prior Change by Distribution Matching

$CE(P,Q) = -PLog_2(Q) - (1-P)Log_2(1-Q)$

Figura 15: Inventario de métricas de evaluación Cuantificación II.

Este segundo grupo de medidas incrementa el grado de sofisticación frente a las anteriores:

- **Entropía normalizada**: Forman en [For07] propone como alternativa al error relativo la utilización de la entropía cruzada normalizada. La normalización permite asignar una medida cero a una predicción perfecta.
- **Divergencia de Kullback-Leibler (KL)**: la divergencia de Kullback-Leibler es una medida muy popular ya que se adapta naturalmente tanto a problemas de clasificación binaria como de multiclasificación. La métrica toma valores en la recta real, alcanzando el valor óptimo en el 0. Su principal ventaja tal como mencionan González et al. en [Gon+17] radica en que los promedios de la medida sobre un diferentes conjuntos de test es más adecuado. Tampoco está definida si alguna de las clases tiene probabilidad cero. En ese caso se utiliza una versión normalizada tal como muestran Gao y Sebastiani en [GS15].

$$NKL = 2\frac{e^{KL(P,Q)}}{1 + e^{KL(P,Q)}} - 1$$

de 0.30 que en el caso de 0.02

Con el fin de dotar de un marco general en el que analizar las distintas medidas Sebastiani en [Seb18a] establece una colección de propiedades que deberían cumplir las medidas así como una taxonomía de las medidas en función de estas propiedades.

2.2. Ensembles

En el aprendizaje automático los *Ensembles* constituyen una familia de métodos que a partir de una colección de modelos base, $\mathcal{G} = \{g_1, ..., g_N\}$, estimados bajo determinadas condiciones, generan un modelo G=$G(g_1, ..., g_N)$ aglutinándolos mediante algún tipo de criterio o algoritmo. Las distintas formas de combinación cubren un amplio espectro de posibilidades como la combinación convexa de las salidas, la construcción de los modelos con remuestreo, la selección dinámica...
Los *Ensembles* han sido utilizados tanto en tareas supervisadas como en no-supervisadas, aunque es en las primeras, y sobre todo en la *Clasificación* en donde más prolífica e intensiva ha sido su aplicación. Dado que nuestro interés pasa por su aplicación en los métodos agregativos de *Cuantificación* a partir de ahora particularizaremos nuestra exposición asumiendo que estamos ante clasificadores.

Los aspectos que han espoleado el desarrollo de estas metodologías son:

- Han logrado batir a los modelos de clasificación mas tradicionales mejorando el desempeño de cada uno de ellos.

- Esta mejora se consigue de forma relativamente barata, ya que como han demostrado Hansen y Salamon en [HS90] una condición necesaria y suficiente, en el ámbito de la clasficación, para que el modelo agregado mejore a cada uno de los modelos individuales es que cada uno de ellos mejore al clasificador trivial (aleatorio).

La característica definitoria de los *Ensemble* los contrapone frente a los métodos más tradicionales. Mientras que éstos seleccionan un único modelo del espacio de hipótesis (el óptimo según la medida de error seleccionada), aquellos construyen diversos candidatos sin desechar ninguno de primeras para con posterioridad combinarlos. Esta diferencia formal entre ambos enfoques tiene sus implicaciones materiales. La disposición de una panoplia de modelos reduce las debilidades inherentes a la disposición de un único modelo tal como exponen Dietterich como Zhou en [Die00] y [Zho12] y que a continuación se resumen:

- **Reducción de la varianza del error de predicción.** En el espacio de hipótesis el conjunto de modelos que optimizan la medida de error definida puede ser númeroso si el conjunto de datos es pequeño frente al espacio de hipótesis. La combinación de todos ellos permite reducir el error que supondría quedarse con un único modelo. Esta razón está relacionada con la reducción de la varianza en el error de predicción[21].

[21] Bias-Variance tradeoff.

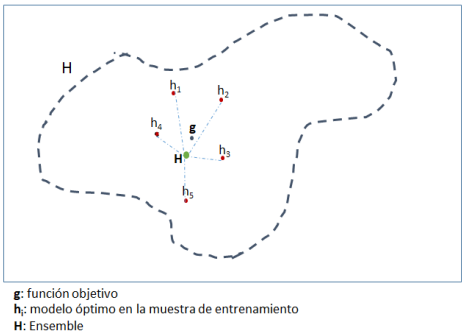

Figura 16: Ejemplo reducción error varianza.

- **Razones computacionales**. Dado que muchos algoritmos de selección realizan una búsqueda de carácter local existe el riesgo de quedar atrapado en un óptimo local. La disposición de más de un modelo permite reducir esta posibilidad.
- **Ventajas representacionales**. Pudiera ser que el modelo subyacente no tuviera representación en el espacio de hipótesis elegido y por tanto el error de predicción fuera elevado. La combinación permite expandir el espacio de representación y aumentar las posibilidades de encontrar el modelo subyacente. Esta relacionado con el concepto de sesgo.

Para el éxito en la generación de un *Ensemble* se requiere que se cumplan al menos dos principios, a saber:

- **Diversidad**. Los modelos base deben ser lo suficientemente diversos para que puedan cubrir ampliamente el espacio de hipótesis, por tanto los conjuntos de entrenamiento sobre los que están construidos deben ser bastante heterogéneos entre sí de cara a generar esa diversidad. Esto es bastante difícil en la práctica dado que sólo se dispone de un conjunto de entrenamiento, por ello técnicas como el *Bagging*, *Boosting* ayudan a soslayar el problema.
- **Precisión**. Los modelos elegidos deben ser poco más predictivos que un modelo aleatorio; se trata, por tanto, de una condición débil y que posibilita que aunque los modelos base sean poco predictivos el modelo final supere a cada uno de los modelos individualmente.

No existe una taxonomía estándar en los *Ensembles* ya que el desarrollo de un algoritmo de esta tipología implica la manipulación de diversos aspectos como son los siguientes:

- **Familia de los clasificadores**. Existen dos alternativas *Ensembles* heterogéneos, en los que se mezclan diversas tipologías de clasificadores, es decir pertenecen a diferentes familias de funciones, u homogéneos. En nuestro caso nos circunscribiremos al caso homogéneo.

- **Funcion de combinación**. Existen diversas alternativas como se verá con posterioridad, las que agregan las salidas de todos los clasificadores (Fusión) a través de algún agregador o las que seleccionan sólo uno de ellos (Selección).
- **Conjunto de entrenamiento**. La batería de clasificadores se genera a partir de diversos subconjuntos de entrenamiento generados a partir del original.
- **Conjunto de Features**. La batería de clasificadores se genera variando el conjunto de features.
- **Target**. La batería de clasificadores se genera agrupando las clases de la variable Target.
- **Incorporación de aleatoriedad**. Una última opción consiste en la incorporación de aleatoriedad en las features consideradas.

Bajo la manipulación de estos elementos se puede establecer una taxonomía como la llevada a cabo por Matteo Re y Giorgio Valentini en [RV12] y que es la que presentaremos a continuación. Esta clasificación parte de dos grandes bloques, a saber, aquellas metodologías que trabajan sobre la función de combinación (*Ensembles No generativos*) manteniendo el resto de parámetros considerados inalterado, y los *Ensemble Generativos* que operan sobre el resto de parámetros:

- *Ensembles No Generativos*. Bajo esta familia se aglutinan todas aquellas variantes que unicamante operan sobre la función de combinación. Se establecen dos posibilidades, por un lado la **Selección** en la que se selecciona un único elemento de la colección de modelos y otra la **Fusión** en la que se combinan todos los clasificadores mediante algún tipo de agregación.
- *Ensembles Generativos*. De forma alternativa están las técnicas **Generativas**, en las cuales el *Ensemble* interviene directamente tanto en el diseño de los modelos base como en la posterior combinación de éstos. Las posibilidades dentro de esta categoría se amplían y se muestran en la figura 17.

Con estos criterios obtenemos el árbol de clasificación mostrado en la figura 17 en el cual incorporamos tanto la clasificación anterior como algunos de sus representantes más importantes:

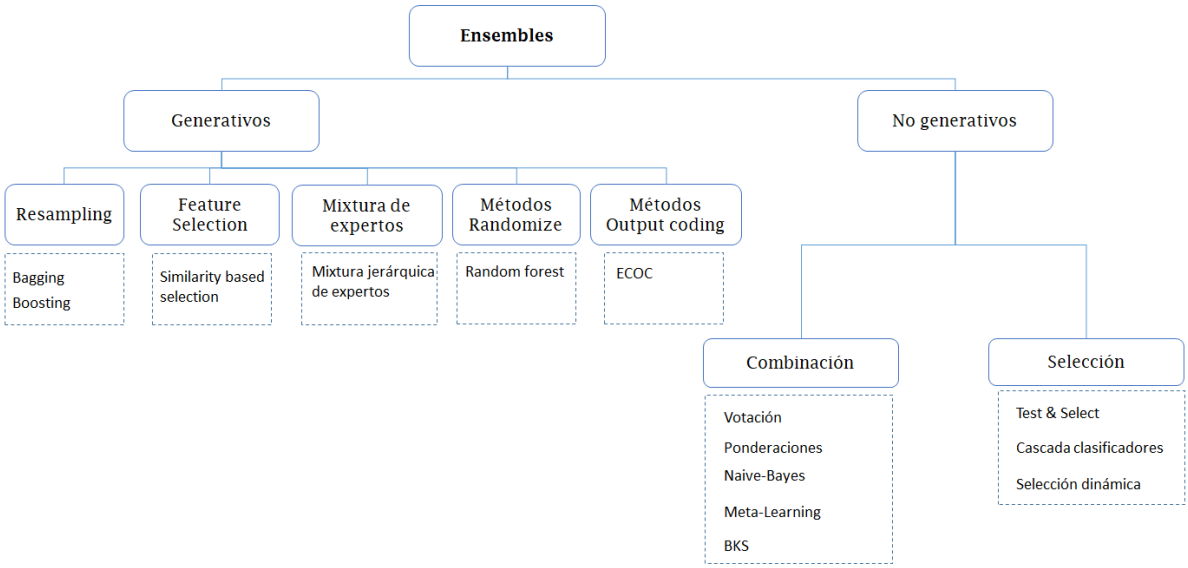

Figura 17: Clasificación de metodologías *Ensemble*.

En los siguientes subepígrafes desarrollaremos las principales categorías mostradas.

2.2.1. Algoritmos No generativos

En esta clase de *Ensembles* la batería de modelos, $\mathcal{G}$, está predeterminada y tan sólo se actuará en la forma que se combinan. Se aprecian dos posibilidades:

- **Combinación por fusión**. Aglutina todos los algoritmos en los que la batería completa de modelos, $\mathcal{G}$, interviene en la generación del output del *Ensemble* mediante algún tipo de agregación. Las más comunes son las presentadas a continuación:
  - **Votación**: Cuando la colección $\mathcal{G}$ está integrada por modelos crisp la combinación más usual es la de la mayoría en sus dos vertientes:
    1. Mayoría absoluta. La clase elegida es aquella que obtiene más de la mitad de los votos. En esta mayoría puede llegar a darse el caso que el *Ensemble* no arroje ningún resultado dado que no se alcance el umbral establecido.

$$G(x) = \begin{cases} c_j, & \text{si } \sum_{i=1}^{N} g_i^j(x) > \frac{1}{2} \sum_{k=1 k\neq j}^{n} \sum_{i=1}^{N} g_i^k(x) \\ \emptyset, & \text{resto de casos} \end{cases} \tag{4}$$

2. Mayoría relativa. Se corresponde con aquella categoría que alcanza un mayor número de votos.

$$G(x) = argmax_j\{\sum_{i=1}^{N} g_i^j(x)\}$$

En los casos mostrados se han equiponderado los votos de cada uno de los modelos, sin embargo nada prohibe que a su vez exista una ponderación, w_i sobre cada uno de los votos. En este caso la ponderación óptima será tal que verificase la siguiente relación de proporcionalidad:

$$w_i \propto log(\frac{e_i}{1-e_i})$$

donde e_i es una medida de la precisión de los modelos, con lo que en el caso de introducir ponderaciones en el sistema de votaciones están deben ser proporcionales al log-odds de la precisión.

- **Ponderaciones**. Cuando $\mathcal{G}$ contiene modelos soft la combinación por ponderación consiste en realizar una combinación lineal de éstos:

$$G(x_1, ..., x_N) = \sum_{i=1}^{N} \omega_i x_i$$

Si todos los pesos ω_i sin iguales estaremos ante el promedio simple, mientras que si son desiguales surge la cuestión de qué elección resultaría adecuada, en este sentido Perrone y Cooper en 1993 demostraron que la elección óptima de ponderaciones viene dada por:

$$\omega_i = \frac{\sum_{j=1}^{T} C_{ij}^{-1}}{\sum_{k=1}^{T}\sum_{j=1}^{T} C_{kj}^{-1}}$$

donde:

$$C_{ij} = \int (g_i(x) - Y)(g_j(x) - Y)dP(y, x)$$

donde h_i se corresponde con cada uno de los modelos e Y es la variable objetivo.

- **Naive-Bayes**. En el caso de que la colección de modelos disponible $\mathcal{G}$ pudiese ser interpretada como una colección de ditribuciones de probabilidad[22] ($g(x) = P(y|x)$) se podría considerar una perspectiva Bayesiana que permitiese generar el *Ensemble* como la esperanza a posteriori:

$$G(x) = \sum_{g\in\mathcal{G}} P(y|g, x)P(g) = \sum_{g\in\mathcal{G}} h(x)P(g)$$

[22] Por ejemplo en el caso de disponer de clasificadores soft.

donde $P(g)$ es la probabilidad de cada hipótesis g.

- **Funciones de meta-aprendizaje**: Esta última forma de combinación consiste en entrenar un modelo que sirva de agregador de los modelos más simples. Es decir cada uno de los g en $\mathcal{G}$ sirven como entrada al modelo del segundo nivel denominado meta-modelo.
- **BKS**: Esta última forma de combinación consiste en entrenar un modelo que sirva de agregador de los modelos más simples. Es decir cada uno de los g en $\mathcal{G}$ sirven como entrada al modelo del segundo nivel denominado meta-modelo.

- **Combinación por selección.** Están formadas por todos aquellos algoritmos en los que para cada instancia se elige como output aquel "mejor"modelo base. Para llevar a cabo esta combinación se debe diseñar como valorar la valía de cada modelo en función de la instancia que esté valorando así como el criterio de selección.

2.2.2. Algoritmos Generativos

Los *Ensembles* generativos estan formados por aquellos en los que se actua tanto sobre el diseño de los modelos base (por ejemplo, alterando el conjunto sobre el que se entrenan, modificandoles) como en las reglas de selección. Siguiendo la taxonomía se han identificados varias categorías:

- **Resampling methods**. Engloban todos los métodos en los cuales el conjunto de modelos base, $\mathcal{G}$ es entrenado sobre conjuntos de información que han sido modificados bajo algún criterio. Bajo esta categoría se engloban dos de los más populares *Ensembles*: Boosting y Bagging.
  - Bagging. El bagging construye la colección de modelos $\mathcal{G}$ a partir de diferentes conjuntos de entrenamiento construidos mediante remuestreo del conjunto total, es decir aplicando Bootstrapping. El objetivo del bagging es poder generar la mayor cantidad de modelos base independientes. Para alcanzar un elevado grado de independencia entre los distintos modelos se generan muestras que deben tener un elevado grado de heterogeneidad entre sí, cuanto más diferentes sean estas muestras mayor será la diversidad de los modelos base y por tanto mejor funcionará el *Ensemble*.

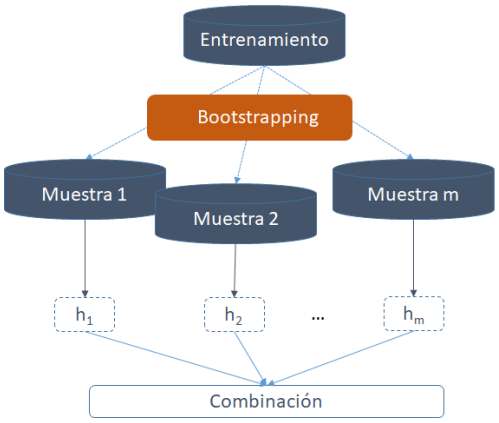

Figura 18: Esquema general del Bagging.

- Boosting. Al contrario que el Bagging en el Boosting los modelos base obtenidos son dependientes entre sí, ya que los clasificadores que se van construyendo dependen entre sí. El caso paradigmático es cuando las distintas muestras, sobre las que se generan los modelos, se construyen con distribuciones que sobrerrepresentan los elementos en los que se cometen errores.

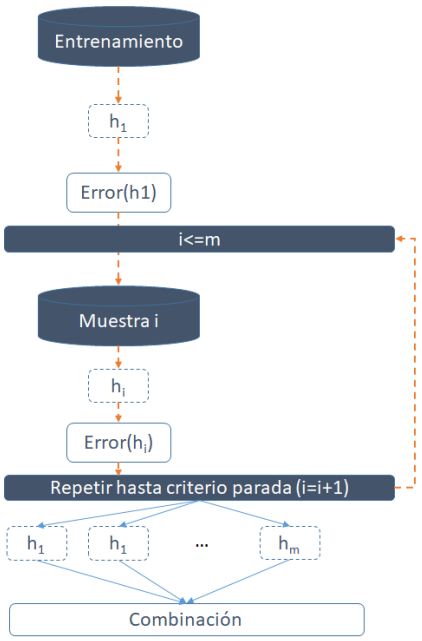

Figura 19: Esquema general del Boosting.

En la figura 19 vemos que en cada una de las iteraciones que se producen las nuevas muestras éstas son generadas con una distribución probabilística que tiene en cuenta que los errores cometidos en el paso previo, ponderando más aquellas observaciones en las que el modelo no predice adecuadamente. Este es el caso del método AdaBoost, planteado por Freund and Schapire en 1995 en el trabajo [FS97], que es uno de los algoritmos pioneros dentro de los englobados bajo la etiqueta *Boosting*. En este caso para generar las diferentes muestras, sobre las que se construye cada uno de los clasificadores, h_t, se usa una distribución exponencial del error cometido en la muestra anterior:

$$\mathcal{P}_t(x) = \begin{cases} e^{\alpha_t}, & \text{si } h_t(x) \neq f(x) \\ e^{-\alpha_t}, & \text{si } h_t(x) = f(x) \end{cases}$$

donde α_t es una función del error,e_t , cometido en la iteración anterior:

$$\alpha_t = \frac{1}{2} ln(\frac{1 - e_t}{e_t})$$

Otra de las variedades más representativa de los *Boosting* son los *Gradient Boosting*, del que el Xgboost se erige como su máximo representante a la vez que uno de los que mejores resultados, en términos de precisión, tienen en aplicaciones prácticas a día de hoy.

Bajo el término Xgboost se engloban diversas variantes de implementaciones y no existe una versión canónica, si bien las diferencias entre una y otras implementaciones difieren en aspectos, que si bien son importantes[23], no impiden entender la lógica general de funcionamiento. En principio todos estos algoritmos tienen como función G una combinación lineal de los clasificadores:

$$G(h_1, ..., h_m) = \sum_{i=1}^{m} \beta_i h_i$$

La determinación de cada h_i se realiza minimizando iterativamente una función de pérdida, L, que dependen de la muestra $\mathcal{S} = \{x_i, y_i\}_{i=1}^{N}$ y de las derivadas primeras y segundas de :

$$\hat{\phi}_m = \arg\min_{\phi \in \mathcal{H}} \sum_{i=1}^{N} \frac{1}{2} \hat{H}_m(x_i) \left[-\frac{\hat{F}_m(x_i)}{\hat{H}_m(x_i)} - \phi(x_i) \right]^2$$
$$\hat{h}_m(x) = \alpha \hat{\phi}_m(x).$$

donde α es la tasa de aprendizaje y $\hat{F}_m$, $\hat{H}_m$ representan el gradiente

[23] Inclusión de términos de regularización, paralelización...

y el hessiano de la función de error L evaluada en la iteración anterior:

$$\hat{F}_m(x_i) = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=\hat{h}_{(m-1)}(x)}$$
$$\hat{H}_m(x_i) = \left[\frac{\partial^2 L(y_i, f(x_i))}{\partial f(x_i)^2}\right]_{f(x)=\hat{h}_{(m-1)}(x)}$$

En los estudios que haremos a lo largo del documento utilizaremos tanto el Adaboost como una implementación del Xgboost adaptada a problemas de clasificación.

- **Feature Selection.** En vez de operar sobre las observaciones de las muestra para generar nuevas submuestras los métodos de selección de features seleccionan subconjuntos de features para generar una variedad de modelos que difieran en el conjunto de predictores que utilizan.

- **Mixtura de expertos.** Este tipo de *Ensembles* están muy relacionados con las redes neuronales y se basan en el parcelamiento del espacio de features para que cada uno de los modelos base se especialice en dicha parcela. Una red, denominada *gating network*, es la responsable de realizar el parcelamiento y de asignar una medida de confianza (o probabilidad) a cada uno de los modelos en cada parcela. Finalmente mediante algun tipo de combinación se genera la salida final.

- **Métodos Randomize.** Los métodos randomize intentan generar modelos base muy diversos actuando en diversos aspectos para conseguir dicha heterogeneidad:

  - Muestreo. De igual forma a como se hace en el Bagging se generan mediante remuestreo multiples muestras sobre las que se estiman los modelos base.
  - Features. Con un enfoque similar al de los *Feature Selection* se incorporan para cada una de las muestras generadas conjuntos diferentes de features.
  - Aleatoriedad. Esta acción consiste en incluir perturbaciones en algunos de los inputs a través de alguna distribución (habitualmente mediante gaussianas).

  El algoritmo más conocido dentro de esta categoría son los Random Forest que pueden incorporar la aleatoriedad tanto en el muestreo como en las features, o en ambas.

- **Métodos Output coding** Este tipo de métodos son utilizados en los problemas multiclase y permiten traducirlos a problemas binarios. A diferencia de los métodos habituales utilizados en los problemas multiclase *1 vs Resto* y *1 vs 1* que ofrecen una solución similar al dividir un problema

de clasificación de clases múltiples en un número fijo de problemas de clasificación binaria, estos métodos permite que cada clase se codifique como un número arbitrario de problemas de clasificación binaria. Cuando se usa una representación *sobredeterminada*, permite que los modelos adicionales actúen como predicciones de "corrección de errores" que pueden resultar en un mejor desempeño predictivo.

3. Ensembles y Cuantificación

Al inicio del documento señalabamos que uno de los aspectos en los que la tarea de *Cuantificación* difiere de la tarea de *Clasificación* es en la constancia de la distribución que genera los datos, esta variación desaconsejaba la utilización directa de un clasificador para la resolución del problema.

Por otro lado, uno de los aspectos que señalamos como satisfactorio de los *Ensembles* es la generación de toda una diversidad de modelos con la finalidad de aumentar la cobertura en el espacio de hipótesis. A la vista de estos dos hechos pudiera existir cierta complementariedad entre la tarea de *Cuantificación* y los algoritmos *Ensemble*. El análisis de esta complementariedad, y más concretamente el análisis del rendimiento de las técnicas *Ensemble* frente a las metodologías tradicionales de la *Cuantificación*, es el fin que guiará las siguientes partes expositivas del documento. En resumen, pretendemos responder a la siguiente cuestión: **¿pueden las metodologías *Ensemble* mejorar los métodos agregativos usados en la *Cuantificación*?** Para responder a esta cuestión se han seleccionado tres metodologías *Ensemble*, las más representativas, que serán comparadas con los métodos agregativos, descritos en el epígrafe anterior, para una amplia variedad de conjunto de datos. Antes de abordar esta parte comenzaremos con la revisión de los estudios en los que se analizan los *Ensembles* y la *Cuantificación*.

3.1. Revisión bibliográfica

La utilización de los *Ensembles* en los problemas de *Cuantificación* se ha ido incorporando, si bien con bastante timidez, dentro del acervo de las metodologías usadas en la *Cuantificación*. La literatura existente sobre este particular no es extensa, con los criterios de busqueda aplicados, se han encontrado 3 trabajos:

Trabajo	Autores	Año
Quantification trees.	Letizi Milli, Ana Monreale, Giulio Rossetti, Fosca Giannotti, Dino Pedreschi, Fabrizio Sebastiani	2013
Using ensembles for problems with characterizable changes in data distribution: A case study on quantification.	Pablo Pérez-Gállego, José Ramón Quevedo, Juan José del Coz	2017
Dynamic ensemble selection for quantification tasks.	Pablo Pérez-Gállego, José Ramón Quevedo, Juan José del Coz	2018

Figura 20: Literatura existente ensembles utilizados para Cuantificación.

En todos ellos se trabaja con métodos de *Cuantificación* agregativos y están enfocados a los problemas binarios.

A continuación presentamos un breve bosquejo de cada uno de ellos.

3.1.1. Quantification trees.

En puridad, y en términos cronológicos, la primera aproximación a los *Ensembles* se llevó a cabo en este artículo de Milli et al. [Mil+13] que ya habíamos bosquejado en el epígrafe de cuantificadores. En este artículo, que abordaba el desarrollo de árboles de decisión enfocados a la *Cuantificación*, se presentaba como trabajo adicional el planteamiento de un *Random Forest* con los árboles de *Cuantificación*.
El *Random Forest* diseñado incorpora dos niveles de aleatoriedad, por un lado, la generación de muestras de entrenamiento que se llevan a cabo por *Bootstrapping* y por otro la selección de los rasgos que intervienen en cada árbol.

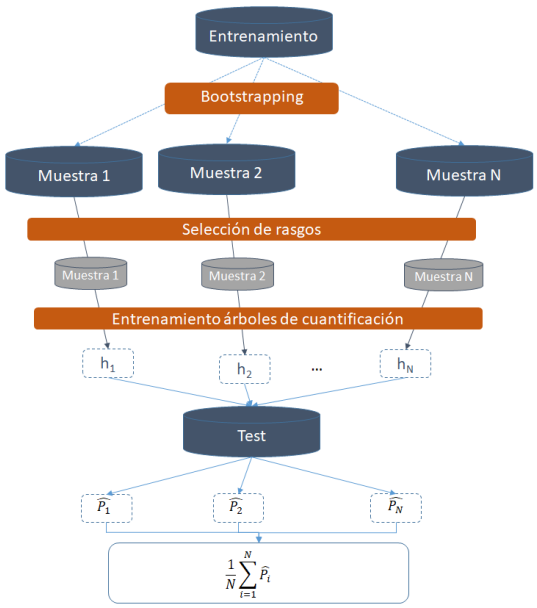

Figura 21: Random Forest de Cuantificación.

Como se puede observar en el esquema superior en la primera etapa se generan las diferentes muestras con las que se entrenaran los distintos árboles, mientras que en la segunda etapa en cada una de ellas se selecciona un conjunto de rasgos con los que se lleva a cabo la estimación individual. La combinación elegida es el promedio simple.
El diseño experimental seguido consiste en evaluar tanto los árboles como los *Random Forest* orientados a la *Cuantificación* sobre tres conjuntos de datos con diferentes niveles de prevalencia, tanto en el conjunto de entrenamiento como en el de test, bajo la métrica KL (Kullback-Leibler). En la mayor parte de los experimentos llevados a cabo, 16 de 18, la metodología *Random Forest*, en sus dos versiones, arroja mejores resultados que las alternativas.

3.1.2. Using ensembles for problems with characterizable changes in data distribution: A case study on quantification.

Este trabajo de 2017 llevado a cabo por Peréz-Gallego, Quevedo y del Coz constituye el primer trabajo, y único hasta ese momento, en el que se aborda de forma monográfica la incorporación de los *Ensemble* en la *Cuantificación* incorporando un marco de experimentación. El diseño experimental llevado a cabo incorpora tres ejes:

- **Metodología de Clasificador**. Se utilizan tres tipos de clasificadores: SVM, Naive Bayes y Regresión logística.
- **Metodología de *Cuantificación***. Se aplican tres técnicas: CC (Clasify and Count), CCA (Clasify and Count Adjusted) y MA (Mixtura agregativa).
- **Metodología *Ensemble.*** Se valoran tres alternativas: dos basadas en métodos *Ensemble* y una utilizando un único clasificador. Para las basadas en *Ensemble* se utilizan dos enfoques:

    1. *Baggings Tradicional*: se entrena un clasificador *Baggins* sobre el conjunto de entrenamiento original. Es decir estaríamos en un método agregativo tradicional en el que sustituimos un clasificador por un *Emsemble*.
    2. *Ensemble Cuantificación*: se trata del punto más novedoso del artículo ya que construye un explícitamente un *Ensemble* a partir de la combinación de diferentes clasificadores entrenados con diferentes niveles de prevalencia. Cada uno de estos clasificadores será transformado en un cuantificador dependiendo de la metodología de *Cuantificación* elegida. Finalmente seran combinados mediante promedio simple.

En el análisis comparativo se ha puesto especial énfasis en comparar para un mismo clasificador y una misma técnica de *Cuantificación* las tres metodologías *Ensemble* propuestas. De esta manera se consigue llevar a cabo una comparativa homogénea entre las técnicas *Ensemble* y la opción individual aislando efectos ajenos. Para la realización del experimento se han utilizado un total de 32 conjuntos de datos.

El procedimiento experimental seguido cubre las siguientes etapas:

1. **Generación de muestras de entrenamiento**. El objetivo de esta etapa inicial es generar conjuntos de entrenamiento con los que entrenar los diferentes clasificadores. En el caso del *Ensemble Cuantificación* estas muestras se generan con diferentes niveles de prevalencia simulados previamente. De esta manera el *Ensemble* dispone de suficientes diversidad en los clasificadores que le conforman al estar entrenados con diferentes niveles de probabilidad de la clase positiva.

2. **Entrenamiento de clasificadores**. En el caso del *Ensemble Cuantificación* para cada muestra de entrenamiento generada en el paso anterior se lleva a cabo la estimación de los clasificadores base. Las otras alternativas llevan a cabo el entrenamiento del clasificador sobre el conjunto original. Se han probado las tres familias de clasificadores mencionadas.

3. **Cuantificación**. En esta etapa se llevan los ajustes necesarios para transformar los clasificadores en cuantificadores. En concreto está orienteada a la determinación de los elementos que intervienen en el método CCA (TPR y FPR), ya que el resto de metodologías propuestas o no requieren de ajuste (CC) o bien sólo cobran sentido en su aplicación (MA).

4. **Evaluación**. En esta etapa se da inicio a la fase test. La construcción de la muestra de test se lleva a cabo con un proceso análogo al de la validación cruzada, en este caso se utiliza 2xCV5, con la salvedad de que sobre la *fold* de test se generan artificialmente con diferentes niveles de prevalencia de 100 muestras de test, es decir se seguiría un esquema como el que se muestra a continuación:

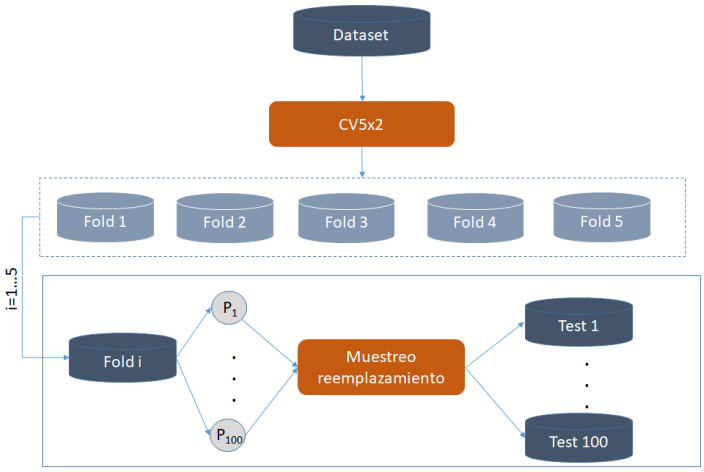

Figura 22: Generación de muestras de Test.

   Con cada una de las muestras de test se procede a la aplicación de cada una de las alternativas que surge de la combinación de los tres ejes determinantes del experimento.

Tanto en los experimentos en los que utiliza como clasificador SVM o Naive Bayes el *Baggings* Modificado arroja mejores resultados que las otras alternativas para todas las posibles técnicas de *Cuantificación*. En el caso de la Regresión Logística y sólo para la *Cuantificación* CC la alternativa del *Baggings* Modificado arroja peores resultados.

3.1.3. Dynamic ensemble selection for quantification tasks.

Se trata de una continuación del trabajo iniciado en el artículo anterior por los mismos autores. Aunque no está enfocado en la construcción de *Ensembles* de cuantificadores se introduce en el análisis de los criterios de selección de los cuantificadores base del *Ensemble*. Es decir, para la familia G=$\{h_1, ..., h_N\}$ de cuantificadores generados se busca algún criterio que permita extraer los k *mejores* $\{h_{i_1}, ..., h_{i_k}\}$.

Se plantean dos alternativas, a saber: alternativa estática y alternativa dinámica:

- **Alternativa estática.** En la alternativa estática la selección del subconjunto de modelos se realiza en la fase de entrenamiento, y por tanto dicho subconjunto es fijo en la aplicación a diversos conjuntos de test. El esquema de funcionamiento sería como el que sigue:

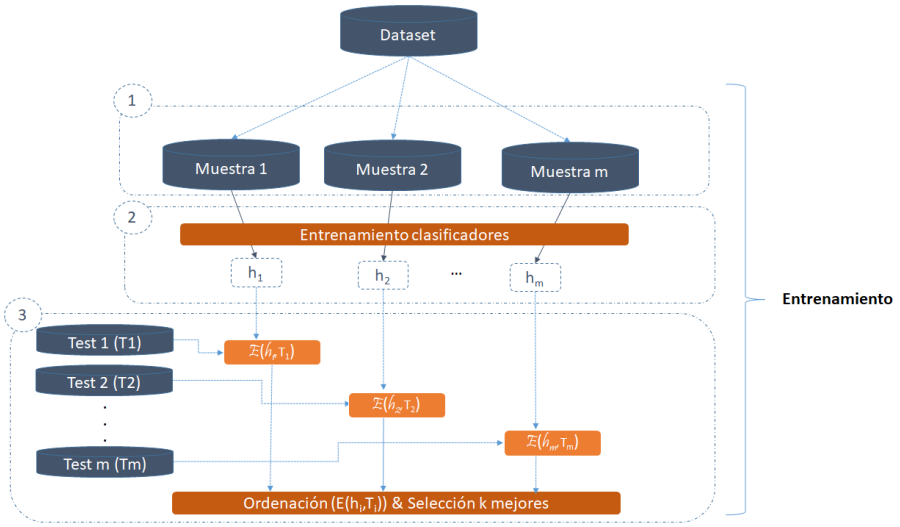

Figura 23: Alternativa estática.

1. **Generación de muestras**. El objetivo de esta etapa inicial es generar conjuntos de entrenamiento con los que entrenar los diferentes clasificadores.
2. **Entrenamiento de clasificadores**. Sobre cada muestra de entrenamiento generada en el paso anterior se lleva a cabo la estimación del clasificador. Únicamente se analizará la Regresión Logística.
3. Esta etapa constituye la parte mollar de la alternativa ya que es donde se lleva a cabo la selección través del orden de mérito establecido por una medida de error. Los autores ensayan dos medidas:
   - ME. El error cuadrático medio.

- MAX. Se trata de una medida basada en el método desarrollado por Forman en [For05] que toma el máximo de la diferencia de las TPR y FPR de cada modelo.

$$MAX(h) = Max_{h \in \mathcal{H}}\{TPR(h) - FPR(h)\}$$

donde $\mathcal{H}$ es la colección de modelos del *Ensemble*.

La medición del error se realiza sobre unos conjuntos de "test"que están construidos a partir las muestras de entrenamiento del resto de modelos, es decir:

$$T_i = \cup_{j \neq i} E_j$$

A partir de la selección de los k mejores modelos se obtiene el *Ensemble* que será de utilización.

- **Alternativa Dinámica**. Al contrario que en el enfoque anterior la alternativa dinámica determina la selección del subconjunto de modelos durante la fase de test, es decir, no existe un orden predefinido sino que este es dinámico. Es de esperar que este tipo de medidas sean las que mejor se adapten a la naturaleza de los problemas ligados a la *Cuantificación*. Su filosofía se basa en la selección de aquellos modelos cuya muestra de desarrollo más se "parezca" a la muestra test que se está evaluando.

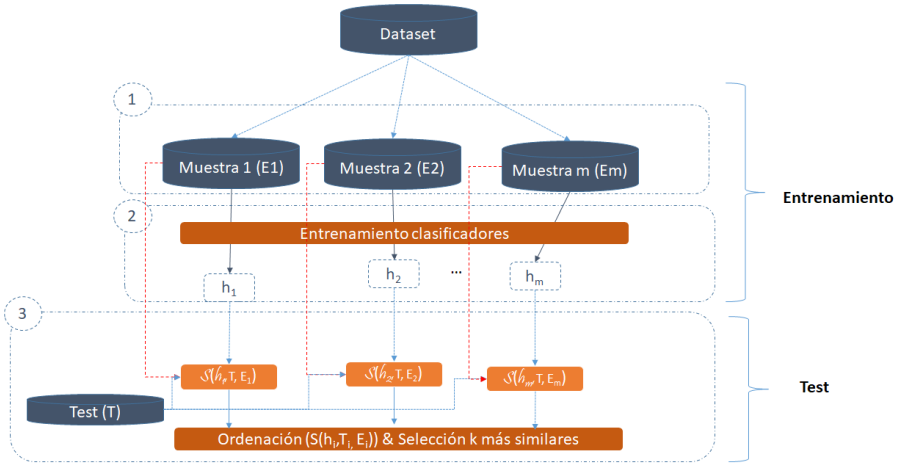

Figura 24: Alternativa Dinámica.

1. **Generación de muestras**. El objetivo de esta etapa inicial es generar conjuntos de entrenamiento con los que entrenar los diferentes clasificadores.
2. **Entrenamiento de clasificadores**. Sobre cada muestra de entrenamiento generada en el paso anterior se lleva a cabo la estimación del

clasificador que al igual que la alternativa estática será la Regresión Logística. Con esto concluye la fase de entrenamiento.

3. En la fase de test es donde se realiza la selección del subconjunto de modelos que formarán parte del *Ensemble* final que se aplicará al conjunto de Test en evaluación. Para ello se calculará la similaridad, S, del conjunto de Test respecto a las diferentes muestras de construcción. Los autores plantean dos medidas:

   - P_{Tr}: Esta medida mide la discrepancia de la frecuencia de la clase en la muestra de test con la frecuencia de la clase en las distintas muestras de desarrollo de los modelos del *Ensemble*. A priori este criterio no es operativo, ya que el objetivo de la *Cuantificación* es la determinación de la frecuencia de las clases en la muestra de test. Para solventar el incoveniente utilizan como frecuencia de la muestra de test la obtenida al aplicar el método de *Cuantificación*, es decir, ordenan según el siguiente criterio:

     $$P_{Tr}(h,i) = d(P_{\mathcal{T}}(i,h), P_{\mathcal{S}}(i,h))$$

     donde d es la medida de error que se considere (ME,AE...), y $P_{\mathcal{T}}(\hat{\imath},h)$ es la estimación que hace el cuantificador h de la frecuencia de la clase i en la muestra $\mathcal{T}$ de test.
   - DS: Esta medida está inspirada en la distancia de Hellinger que los autores utilizaron en el trabajo [GAA13] comparando mediante esta medida, la salidad de cada uno de los modelos sobre la muestra de test con la que arrojaban sobre sus muestras de entrenamiento. Las distancias así obtenidas permiten establecer un ranking entre los distintos modelos que conforman el *Ensemble*. Mientras que la anterior medida únicamente compara dos valores puntuales, la predicción de la clase en las dos muestras, aquí se tiene en cuenta la distribución completa.

   A partir de la selección de los k modelos con las muestras de construcción más similares a la muestra de Test se construye la predicción de la frecuencia de la clase en dicha muestra.

Los resultados obtenidos por los autores muestran mejores resultados en los métodos de selección planteados frente a la alternativa del promedio global.

3.2. Análisis

Tal cómo se apuntaba al inicio de esta sección es significativa la escasez de documentos en los que se estudia la introducción de la metodologías *Ensemble* en los problemas de *Cuantificación*. De los tres documentos sólo uno está enfocado al análisis de la problemática, [PQC17] , mientras que los otros dos lo abordan secundariamente. Analizaremos a continuación, sucintamente, algunos

de los aspectos más relevantes de los documentos revisados.

En lo que respecta a los *Ensemble* utilizados en [PQC17] cabe señalar que son todos del tipo *baggings*, si bien variando el clasificador subyacente; no obstante dada la variedad de familias de *Ensembles* existentes la experimentación con otro tipo de éstos, por ejemplo los tipo *Boosting*, hubiera arrojado nuevas conclusiones en el análisis llevado a cabo. En este sentido el enfoque llevado a cabo en este documento es precisamente explorar la alternativa de los *Boosting* para ver su utilidad en los los problemas de *Cuantificación*. Un aspecto relevante por el cual los métodos Boosting pudieran tener mejor adaptación a los métodos Baggins es en la selección de los pesos que conforman la función de combinación de los modelos subyacentes. Mientras que el caso de un método *Baggins* la combinación de los clasificadores suele ser fijada a discreción por el investigador (en el caso de [PQC17] se ha utilizado una media simple si bien se deja abierta la investigación en esta línea) en los métodos *Boosting*, por ejemplo Adaboost, seleccionan los pesos de forma que tienen en cuenta los errores cometidos por los clasificadores.

Por otro lado, los clasificadores subyacentes utilizados en [PQC17] y en [Pér+18] son algoritmos de clasificación que no están particularizados para la labor de *Cuantificación* al contrario que en el caso de [Mil+13] donde los árboles de decisión han sido adaptados específicamente al problema en cuestión. Este hecho pudiera derivar en que los resultados obtenidos en [PQC17] pudieran haber sido mejorados, sin embargo al ser una vía no explorada no es posible emitir un juicio al respecto. En este sentido nuestro análisis tampoco ha explorado esta vía si bien es una de las vías futuras de continuidad de la investigación.

4. Experimentos

En esta sección se llevará a cabo tanto la presentación de los resultados obtenidos en los experimentos desarrollados como la descripción y objetivos que persiguen éstos. La finalidad de los experimentos diseñados pretende dar respuesta a si la utilización de *Ensembles* puede sustituir la utilización de algunas de las técnicas propias surgidas para dar solución a la *Cuantificación*. Las metodologías *Ensemble* valoradas han sido: Random Forest, AdaBoost y Xgboost. Para cada una de ellas se analiza y confronta su efectividad frente al resto de metodologías de *Cuantificación* descritas con anterioridad. Con estos análisis se cubriran los siguientes objetivos:

1. **Efectividad Relativa Cuantificación**: Comprobar las conclusiones obtenidas en estudios previos sobre la efectividad relativa de las técnicas de *Cuantificación*.

2. **Efectividad Relativa *Ensemble***: Estudiar la efectividad relativa dentro de las técnicas *Ensemble* de las metodologías Boosting (Xgboost y AdaBoost) que hasta el momento no habían sido analizadas en estudios previos.

3. ***Ensemble* vs Método CC**: Comprobar que las técnicas *Ensemble* mejoran las aplicaciones de clasificadores simples (métodos CC) en los problemas de *Cuantificación*.

4. ***Ensemble* vs Otros Métodos**: Estudiar la efectividad relativa de los *Ensemble* frente a las otras técnicas de *Cuantificación* (métodos CCA, EMQ y MA).

5. **Influencia Nivel Prevalencia**: Analizar la influencia de los diferentes niveles de prevalencia en las efectividades de los métodos *Ensemble* y métodos de *Cuantificación*.

6. **Ajuste CCA en Xgboost**: Estudiar si la aplicación del método CCA a los métodos Xgboost permite mejorar los resultados obtenidos sin su aplicación.

Para poder dar respuesta se ha utilizado un variada colección de conjuntos de datos, 17 en total, sobre los que se ha aplicado cada una de las metodologías comentadas.
La estructura de este capítulo está dividido en dos grandes bloques; en el primero se explica el diseño elegido así como los conjuntos de datos y algoritmos, tanto de *Cuantificación* como *Ensemble*, usados. En la segunda se muestran y analizan los resultados sobre la base de los objetivos establecidos.

4.1. Diseño experimental

4.1.1. Conjuntos de datos

Para la evaluación de la hipótesis se han utilizado una colección de conjuntos de datos reales que están disponibles en la web *UCI Machine Learning Repository*[24]. Todos estos conjuntos seleccionados están orientados a problemas de clasificación debido a que no existen conjuntos de datos especializados para la *Cuantificación*. Cada uno de los conjuntos de datos mantiene la siguiente estructura:

- Rasgos. Los ficheros contienen una colección de variables potencialmente predictoras del Target de naturaleza numérica. Durante todo el proceso experimental se han respetado estos rasgos y no se han construido rasgos adicionales.
- Target. Algunos de los ficheros utilizados mantienen variables Target multiclase, para este tipo de situaciones se ha recurrido a la transformación en un problema binario generando tantos ficheros como categorías existan, y en cada uno de ellos manteniendo una categoría frente al resto (*One vs Rest*).

Se han seleccionado 17 conjuntos de datos diferentes cuyas principales características se muestran en las figuras 25 y 26:

[24] https://archive.ics.uci.edu/ml/datasets.php

ID	Nombre Fichero	Fuente	Descripción	Tamaño	Variables explicativas	Target	Prevalencia
D01	Breast_Cancer_Wisconsin	https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29	Conjunto de información recopilado en los hospitales de Wisconsin referentes a historias clínicas de pacientes con cancer de pecho. Cada registro contiene 9 variables explicativas, un identificador de la historia y una variable target en función de la severidad del tumor.	699	9	2: Benigno (-1) 4: Maligno (+1)	35,48%
D02	Iris_Setosa	https://archive.ics.uci.edu/ml/datasets/Iris	Conjunto formado por las medidas de una colección de lirios realizada por el botánico inglés Ronald Fisher en 1936. El conjunto de información consta de 150 observaciones equidistribuidas en las tres categorías de lirios, a saber: Setosa, Versicolor, Virginica. Para su tratamiento se ha transformado el conjunto original en tres conjuntos cada uno de los cuales enfrenta una de las categorías al resto.	150	4	Setosa: (+1) Resto: (-1)	33,33%
D03	Iris_Versicolor			150	4	Versicolor: (+1) Resto: (-1)	33,33%
D04	Iris_Virginica			150	4	Virginica: (+1) Resto: (-1)	33,33%
D05	Spam	https://archive.ics.uci.edu/ml/datasets/Spambase	La base de datos dispone de los registros de eventos de Spam recibidos por correo electrónico.	4.601	57	0: No (-1) 1: Sí (+1)	39,40%
D06	Red_Wine	https://archive.ics.uci.edu/ml/datasets/Wine+Quality	El conjunto de información contiene las valoraciones que realizan catadores de Vino verde portugues además de sus principales propiedades químicas.	1.599	11	3,4,5: Malo (-1) 6,7,8: Bueno (+1)	53,4%
D07	White_Wine			4.898	11	3,4,5: Malo (-1) 6,7,8,9: Bueno (+1)	66,52%
D08	Yeast	https://archive.ics.uci.edu/ml/datasets/Yeast	Conjunto de información que recoge datos para la clasificación de las proteínas en función de su localización.	1.484	8	1: Nuclear (+1) 0: Resto (-1)	28,90%
D09	German_Credit	https://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29	Conjunto de información que recoge impagos de clientes alemanes.	1.000	24	1: Normal (-1) 2: Impago (+1)	30%
D10	Tic-Tac-Toe Endgame	https://archive.ics.uci.edu/ml/datasets/Tic-Tac-Toe+Endgame	Conjunto de datos con las posibles jugadas en el juego de cruz y raya.	958	9	0: Pierde el primer jugador (-1) 1: Gana el primer jugador (+1)	34,67%
D11	Cardiotocography_Patologica	https://archive.ics.uci.edu/ml/datasets/Cardiotocography#	Se procesaron automáticamente 2126 cardiotocogramas fetales (CTG) y se midieron las características de diagnóstico respectivas. Los CTG también fueron clasificados por tres obstetras expertos y una etiqueta de clasificación de consenso asignada a cada uno de ellos. La clasificación hace	2.126	21	1: Patológica (+1) 0: Resto (-1)	8,27%
D12	Cardiotocography_Normal			2.126	21	1: Normal (+1) 0: Resto (-1)	77,84%
D13	Cardiotocography_Sospechosos			2.126	21	1: Sospechoso (+1) 0: Resto (-1)	13,87%
D14	Semeion	https://archive.ics.uci.edu/ml/machine-learning-databases/semeion/	Los datos se extrajeron de imágenes que se tomaron de números escritos a mano por 80 personas de dos formas, una rápida y otra rapida.	1.593	256	1: Digito 2 (+1) 0: Resto (-1)	9,99%
D15	Cmc.1	https://archive.ics.uci.edu/ml/datasets/Contraceptive+Method+Choice	Conjunto de datos que recogen un muestra del estudio de uso de métodos anticonceptivos en Indonesia.	1.473	9	1: No usa anticonceptivos (+1)	57,3%
D16	Cmc.2			1.473	9	1: Usa métodos de LP (+1) 0: Resto (-1)	77,40%
D17	Cmc.3			1.473	9	1: Usa métodos CP (+1) 0: Resto (-1)	65,30%

Figura 25: Conjuntos de información valorados.

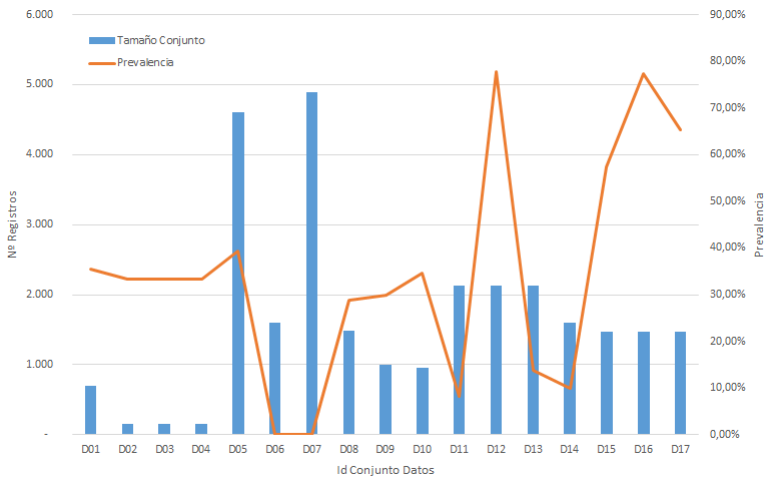

Figura 26: Conjuntos de información valorados.

Como se puede observar los conjuntos manejados abarcan diferentes tamaños

al igual que diferentes niveles de prevalencia, con el fin de analizar el desempeño de los *Ensemble* bajo diferentes configuraciones.

4.1.2. Metodología

La metodología experimental establecida analizará si la introducción de un Ensemble mejora a los métodos clásicos de *Cuantifación* en términos de predicción. Con este fin se ha diseñado el siguiente experimento que esquematizamos en la figura 27:

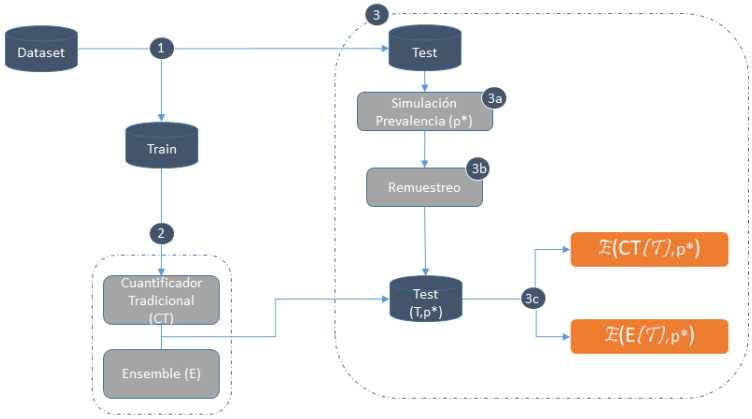

Figura 27: Metodología.

En este diseños se llevan a cabo tres fases diferenciadas:

1. **Etapa 1**. Fijado un dataset se procede escindirlo en dos subconjunto, a saber: uno de Test, $\mathcal{T}$, sobre el que se medirá el desempeño y otro de entrenamiento. La proporción se fijará en un 70 % y 30 % que es un estándar en las técnicas de hold-out.

2. **Etapa 2**. Entrenamiento del cuantificador y el *Ensemble* sobre el conjunto de entrenamiento seleccionado. Tanto para el cuantificador tradicional como para el *Ensemble* se utiliza el mismo conjunto de entrenamiento para la estimación del modelo. De esta forma aseguramos una aplicación homogénea.

   a) Estimación Clasficador. Se tomarán como familias de clasificadores tanto la regresión logística (RL) con regularización como las máquinas de vector soporte con diferentes funciones base (lineal y gaussiana). La determinación de los hiperparámetros de cada una de las familias de algoritmos se realizará mediante validación cruzada de 10

folds. Los hiperparámetros correspondientes a cada familia son los siguientes:

Metodología	Kernel	C	γ
SVM	Lineal	0.001, 0.01, 0.1, 1, 10, 100, 1000	
	RBF	0.001, 0.01, 0.1,1, 10, 100, 1000	0.1, 0.01, 0.001, 0.0001
Regresión Logística	Regresión logisitica	0.001, 0.01, 0.1, 1, 10, 100, 1000	

Figura 28: Parametrizaciones experimentadas para los Clasificadores.

Para las diferentes metodologías *Ensemble* se han escogido los parámetros más relevantes mediante validación cruzada.

Metodología	Parámetro	Rango
Random Forest	Profundidad	3, 6, 10
	Nº Estimadores	100, 500, 1000
XGBoost	Profundidad	3, 6, 10
	Tasa Aprendizaje	0.01, 0.05, 0.01
	Nº Estimadores	100, 500, 1000
AdaBoost	Nº Estimadores	100, 500, 1000

Figura 29: Parametrizaciones de los métodos Ensemble.

b) Aplicación metodologías de *Cuantificación*. Una vez se dispone de los clasificadores (tradicional y *Ensemble*) se realiza la preparación de éstos para la *Cuantificación* aplicando los métodos vistos en el epígrafe 2. Se utilizarán las siguientes metodologías de *Cuantificación*:

- **CC**. No necesita de ningun tipo de manipulación sobre el clasificador ni sobre los *Ensemble*. El resto de los métodos serán aplicados únicamente a los clasificadores, quedando, por tanto, exentos los *Ensemble*.
- **CCA**. Se aplicará al clasificador la corrección prescrita en este enfoque:

$$P(1) = \frac{P_{\mathcal{T}}(\hat{1}) - FPR}{TPR - FPR}$$

  donde $P_{\mathcal{T}}(\hat{1})$ es la probabilidad que surge de la aplicación del clasificador. Las estimaciones tanto del TPR como del FPR serán obtenidas mediante validación cruzada de 10 folds en la muestra de entrenamiento,
- **EMQ**. Se aplicará el algoritmo EM visto en el epígrafe 2.
- **MA**. Se obtendrá la probabilidad de la clase a través de la opti-

mización del siguiente programa:

$$\min_{p} d(F_{\mathcal{T}}, F^1_{\mathcal{S}}p + F^0_{\mathcal{S}}(1-p))$$
$$\text{Sujeto a :}$$
$$0 \geq p \leq 1,\ \forall\, i$$

donde $F_{\mathcal{T}}$ es la distribución de probabilidades en la muestra de test y $F^1_{\mathcal{S}}$ es la distribución de probabilidades en la muestra de entrenamiento para la categoría de interés. Se ha tomado como distancia la de Hellinger.

3. **Etapa 3**. La etapa tres comenienza con la simulación de las diferentes probabilidades de prevalencia (3a). Para cada una de las probabilidades de prevalencia simuladas se generara un conjunto de test mediante remuestreo con reemplazamiento del original que mantenga el nivel de prevalencia simulado (3b). Sobre este conjunto de test se mide el error tanto del cuantificador tradicional como del *Ensemble* (3c) a través de la medida de error MAE[25] (Media del error absoluto entre las dos clases).

   Este ejercicio se repite tantas veces como diferentes simulaciones hayamos hecho del nivel de prevalencia. En este caso se han generado 100 simulaciones de diferentes niveles de prevalencia.

Para llevar a cabo la implementación de este proceso se ha utilizado la librería *QuaPy*[26] de Python desarrollada por A. Moreo, [MES21], y que dispone de todos los métodos apuntados. El entrenamiento de los algoritmos se ha realizado con la librería Sklearn, siendo los módulos utilizados los siguientes:

- Implementación Random Forest: Se utiliza la implementación *ExtraTreeClassifier*.
- Implementación AdaBoost: Se utiliza la implementación *AdaBoostClassifier*.
- Implementación XGBoost: Se utiliza la implementación *GradientBoostingClassifier*.
- Implementación SVM: Se utilizan la implementaciones *SVC* y *LinearSVC*.
- Implementación Regresión Logística: Se utiliza la implementación *LogisticRegression*.

[25] La definición explícita está en la figura 14.
[26] https://hlt-isti.github.io/QuaPy/build/html/index.html

4.2. Resultados

4.2.1. Resultados Generales

Los resultados obtenidos se muestran en las tablas inferiores, una para cada familia de clasificadores (SVM lineal, SVM RBF y regresión logística). En cada una de ellas se muestra para cada conjunto de datos el MAE de cada uno de los métodos de *Cuantificación* y *Ensemble*. Adicionalmente se ha incorporado un ranking[27] que resume la posición de la técnica en función de su precisión en cada uno de los conjuntos evaluados.

Conjunto	Cuantificadores (SVM Lineal)				Ensembles		
	CC	CCA	EMQ	MA	Xgboost	Random Forest	Ada Boost
D01	0,0218	0,0191	0,0202	0,0494	0,0384	0,0119	0,0181
D02	-	-	0,0523	0,0467	-	-	-
D03	0,2577	0,0916	0,1259	0,0640	0,0321	0,0321	0,0321
D04	0,0485	0,0420	0,0698	0,0208	0,0488	0,0642	0,0488
D05	0,0483	0,0100	0,0116	0,0084	0,0230	0,1137	0,0329
D06	0,1353	0,1107	0,0168	0,0439	0,1048	0,1043	0,1096
D07	0,2743	0,0768	0,0266	0,0420	0,1498	0,2272	0,1727
D08	0,1782	0,4298	0,1181	0,0977	0,2210	0,3844	0,2169
D09	0,2994	0,0432	0,0826	0,1275	0,1998	0,2979	0,2238
D10	0,0196	0,0065	0,0110	0,0033	-	0,0147	0,0196
D11	0,0746	0,0322	0,0069	0,0517	0,0179	0,0426	0,0241
D12	0,1340	0,0777	0,0226	0,0867	0,0364	0,0481	0,0412
D13	0,1188	0,0136	0,0122	0,1898	0,0009	0,0464	0,0009
D14	0,1148	0,0214	0,0284	0,1281	0,0745	0,1489	0,1004
D15	0,2234	0,0801	0,0546	0,1145	0,2011	0,1999	0,1726
D16	0,4871	0,5371	0,0402	0,1660	0,4420	0,4111	0,3981
D17	0,4615	0,5001	0,1059	0,1461	0,2877	0,3318	0,3019
Ranking Promedio	**2,059**	**4,059**	**5,235**	**4,412**	**4,235**	**3,118**	**3,941**

Error Mínimo

Figura 30: MAE Promedio SVM Kernel Lineal.

[27] Un mayor ranking implica una mejor capacidad predictiva. La definición del ranking puede consultarse en el anexo.

Conjunto	Cuantificadores (SVM RBF)				Ensembles		
	CC	CCA	EMQ	MA	Xgboost	Random Forest	Ada Boost
D01	0,0232	0,0067	0,0169	0,0446	0,0384	0,0119	0,0181
D02	-	-	0,0503	0,0193	-	-	-
D03	0,0343	0,0477	0,0833	0,0340	0,0321	0,0321	0,0321
D04	0,0495	0,0485	0,0557	0,1309	0,0488	0,0642	0,0488
D05	0,0500	0,0075	0,0144	0,0105	0,0230	0,1137	0,0329
D06	0,1209	0,0332	0,0308	0,0616	0,1048	0,1043	0,1096
D07	0,1954	0,0499	0,0277	0,0311	0,1498	0,2272	0,1727
D08	0,2977	0,1471	0,1262	0,0659	0,2210	0,3844	0,2169
D09	0,2589	0,0611	0,0351	0,0410	0,1998	0,2979	0,2238
D10	0,0347	0,0043	0,0110	0,0063	-	0,0147	0,0196
D11	0,0415	0,0093	0,0224	0,0343	0,0179	0,0426	0,0241
D12	0,0799	0,0232	0,0209	0,1193	0,0364	0,0481	0,0412
D13	0,1085	0,0337	0,0398	0,1871	0,0009	0,0464	0,0009
D14	0,0925	0,0254	0,0318	0,1382	0,0745	0,1489	0,1004
D15	0,1934	0,0722	0,0397	0,0536	0,2011	0,1999	0,1726
D16	0,4175	0,1350	0,0590	0,2840	0,4420	0,4111	0,3981
D17	0,3027	0,2363	0,1062	0,2509	0,2877	0,3318	0,3019
Ranking Promedio	**2,353**	**5,588**	**5,235**	**3,882**	**4,000**	**2,412**	**3,647**

Error Mínimo

Figura 31: MAE Promedio SVM Kernel RBF.

Conjunto	Cuantificadores (Regresión Logística)				Ensembles		
	CC	CCA	EMQ	MA	Xgboost	Random Forest	Ada Boost
D01	0,0521	0,0391	0,0194	0,0533	0,0384	0,0119	0,0181
D02	-	-	0,0114	0,0206	-	-	-
D03	0,3210	0,1930	0,1043	0,1914	0,0321	0,0321	0,0321
D04	0,0607	0,0308	0,0338	0,2552	0,0488	0,0642	0,0488
D05	0,0421	0,0118	0,0052	0,0186	0,0230	0,1137	0,0329
D06	0,1193	0,0604	0,0114	0,0573	0,1048	0,1043	0,1096
D07	0,2428	0,0638	0,0457	0,0409	0,1498	0,2272	0,1727
D08	0,3808	0,0529	0,1640	0,0728	0,2210	0,3844	0,2169
D09	0,2601	0,1246	0,0958	0,1164	0,1998	0,2979	0,2238
D10	0,0196	0,0065	0,0667	0,0027	-	0,0147	0,0196
D11	0,0722	0,0165	0,0096	0,0414	0,0179	0,0426	0,0241
D12	0,1220	0,0523	0,0143	0,1329	0,0364	0,0481	0,0412
D13	0,1086	0,0058	0,0212	0,1432	0,0009	0,0464	0,0009
D14	0,0819	0,0127	0,0045	0,1432	0,0745	0,1489	0,1004
D15	0,1889	0,1274	0,0956	0,1366	0,2011	0,1999	0,1726
D16	0,4337	0,0511	0,0612	0,1780	0,4420	0,4111	0,3981
D17	0,3524	0,1175	0,1056	0,1767	0,2877	0,3318	0,3019
Ranking Promedio	**2,000**	**5,118**	**5,588**	**3,765**	**4,059**	**2,765**	**3,765**

Error Mínimo

Figura 32: MAE Promedio Regresión Logística.

A continuación, y sobre la base de los resultados mostrados, se va a proceder a cubrir cada uno de los objetivos marcados.

4.2.2. Efectividad Relativa de las técnicas de Cuantificación

En lo que respecta a las técnicas de *Cuantificación* se confirma que la metodología CC arroja los peores resultados para todos las tipologías de clasificadores evaluadas en la mayor parte de los conjuntos. Del resto de técnicas (CCA, EMQ y MA) la que implementa el método *Expectation-Maximization*, EMQ, es la que

a lo largo de los tres clasificadores mejor comportamiento tiene, obteniendo el mejor ranking para el caso de la regresión logística y el SVM con kernel lineal. Esto es algo que se ha observado en líneas generales en la bibliografía consultada.

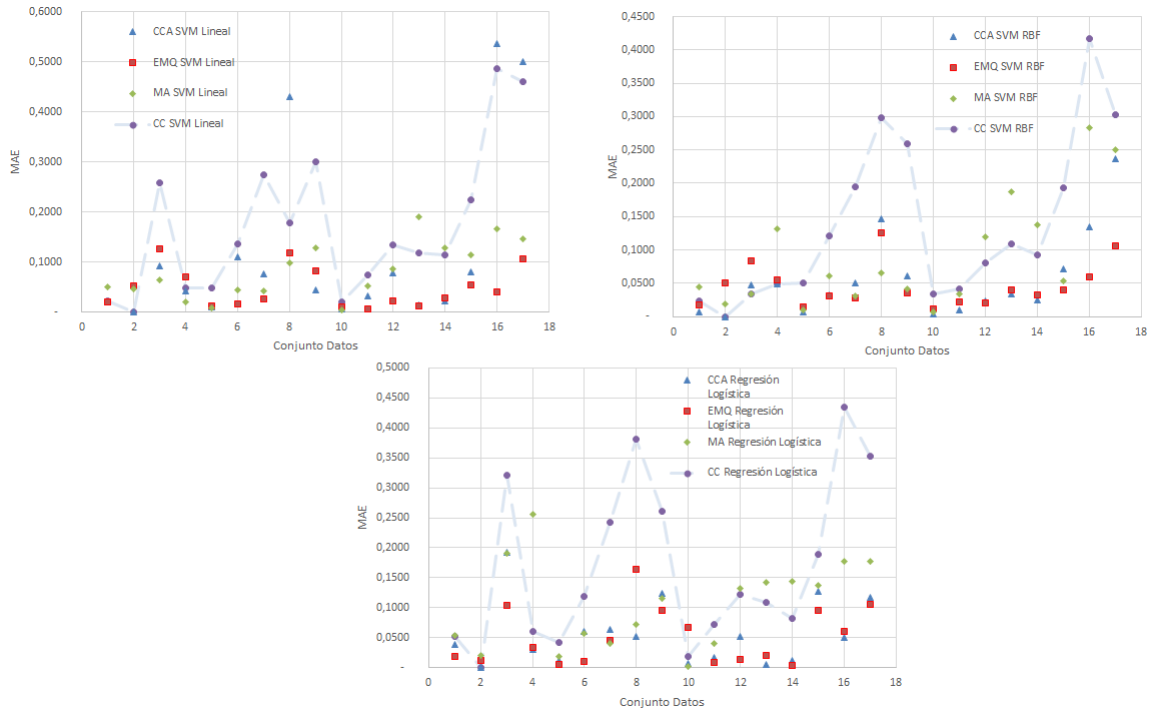

Figura 33: Errores de los métodos de Cuantificación.

4.2.3. Efectividad Relativa de los Ensemble

Desde el punto de vista de los *Ensemble* la metodología XGBoost arroja los mejores resultados en los conjuntos analizados, obteniendo el menor MAE en 12 de los 17 conjuntos evaluados. La metodología AdaBoost alcanza el menor error en los 4 restantes conjuntos siendo el Random Forest el que peor resultados arroja.

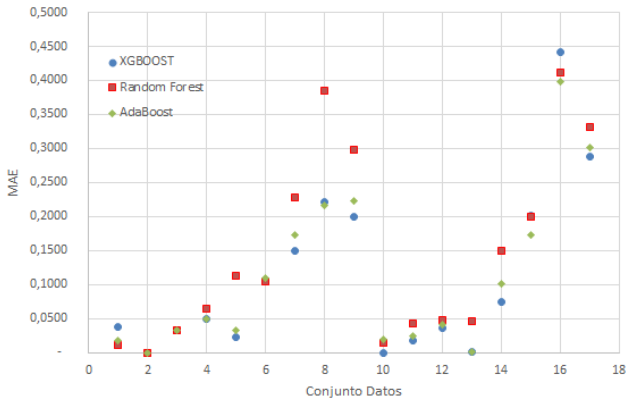

Figura 34: Errores de los Ensemble.

4.2.4. *Ensemble* vs Método CC

En lo que respecta a esta comparativa vemos que las metodologías *Ensemble* baten globalmente a todos los métodos CC, este hecho se confirma obteniendo los ranking de cada uno de los clasificadores evaluados.

	CC SVM Lineal	CC SVM SVM Kernel	CC Reg Logística	XGBOOST	Random Forest	AdaBoost
Ranking	2,06	2,35	2,00	4,10	2,76	3,78

Figura 35: Ranking Ensemble y CC.

Es reseñable la significativa diferencia, en términos de ranking, de la metodología XgBoost frente al resto de alternativas CC. Si analizamos como se comporta en cada uno de los conjuntos el algoritmo XGboost frente al resto de clasificadores se verifica que en la mayor parte de los conjuntos, 12 de 17, obtiene la menor medida de error.

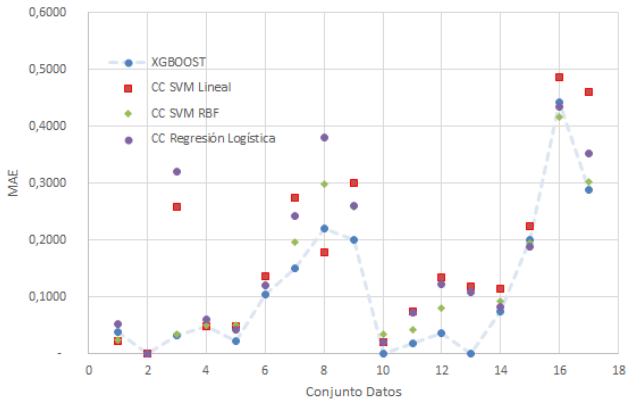

Figura 36: Errores del XGboost frente a CC.

4.2.5. *Ensemble* vs Otros Métodos

Por el contrario frente a los otras técnicas de *Cuantificación* los resultados no son tan optimistas y en la mayor parte de conjunto de datos los resultados son favorable a aquellas.

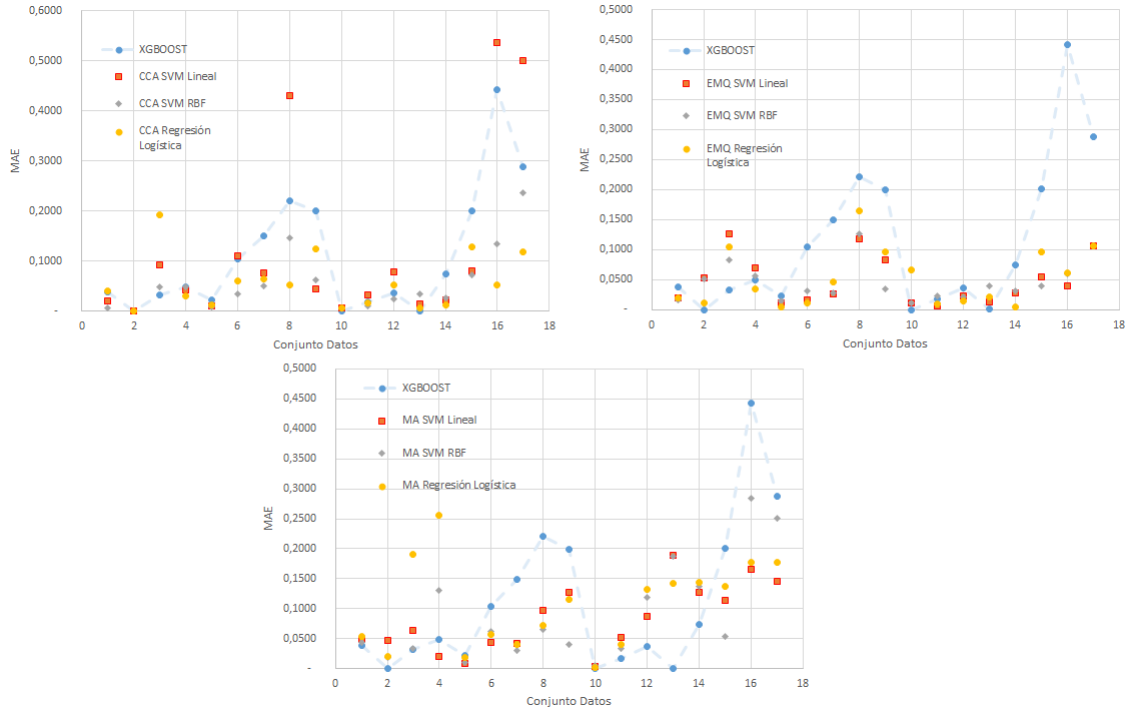

Figura 37: Errores del XGboost frente a resto de alternativas.

Como se puede observar en la mayor parte de los métodos los errores obtenidos por el XGboost, el mejor de los métodos *Ensemble*, son superiores a los que arrojan los métodos CCA, EMQ y MA.

4.2.6. Influencia Nivel Prevalencia

Otro análisis que llevaremos a cabo enfrentará el error del mejor clasificador frente a las tres alternativas de *Ensemble* para cada nivel de prevalencia simulada. En los gráficos que mostraremos a continuación hemos representado en la línea negra discontinua la predicción perfecta y el resto de alternativas han sido representadas mediante líneas interpoladas. De esta manera podemos observar si los errores están concentrados en determinados niveles de prevalencia (p.e.: prevalencias cercanas a uno o cero) a lo largo de todos los conjuntos o por el contrario son dependientes del conjunto de datos que estemos analizando.

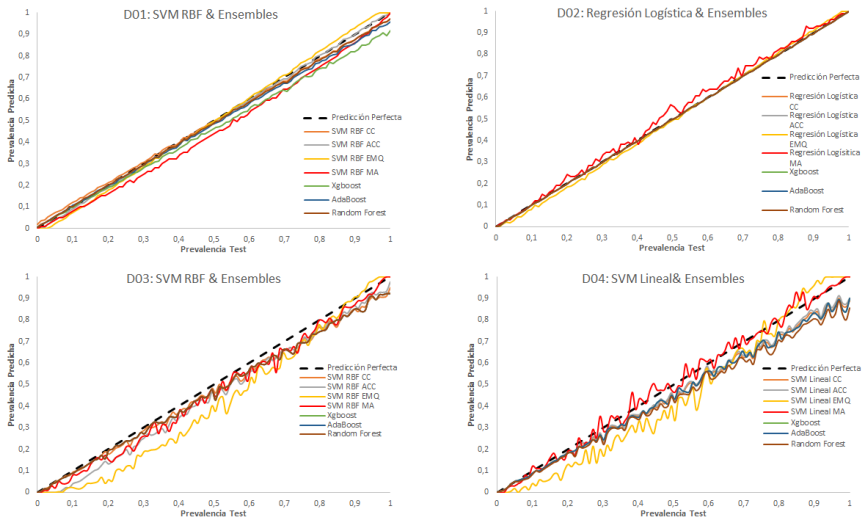

Figura 38: MAE Promedio Regresión Logística.

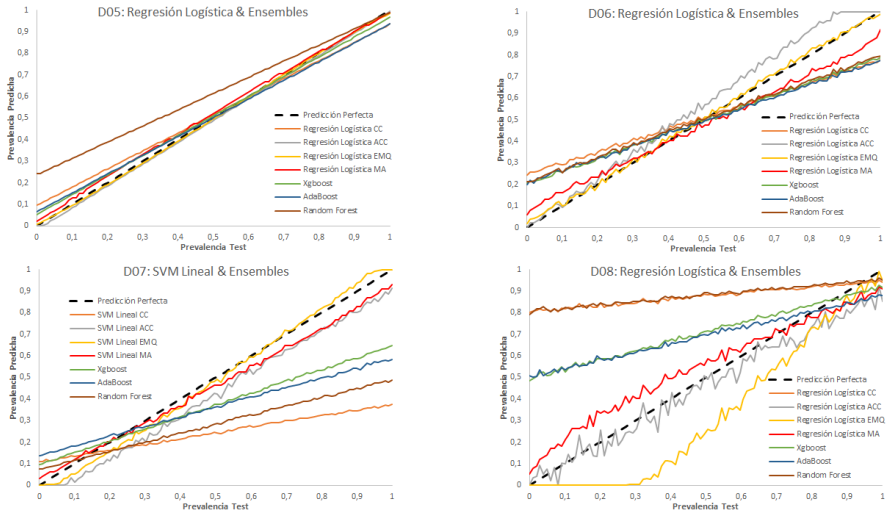

Figura 39: MAE Promedio Regresión Logística.

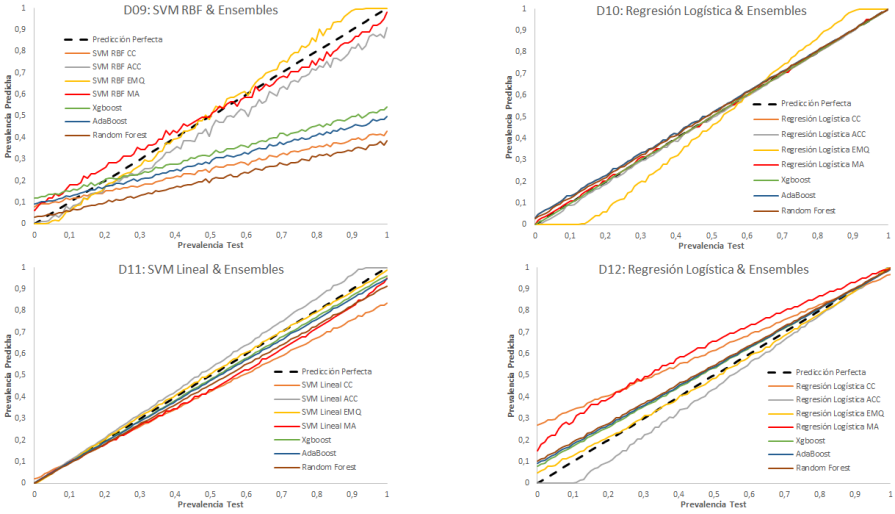

Figura 40: MAE Promedio Regresión Logística.

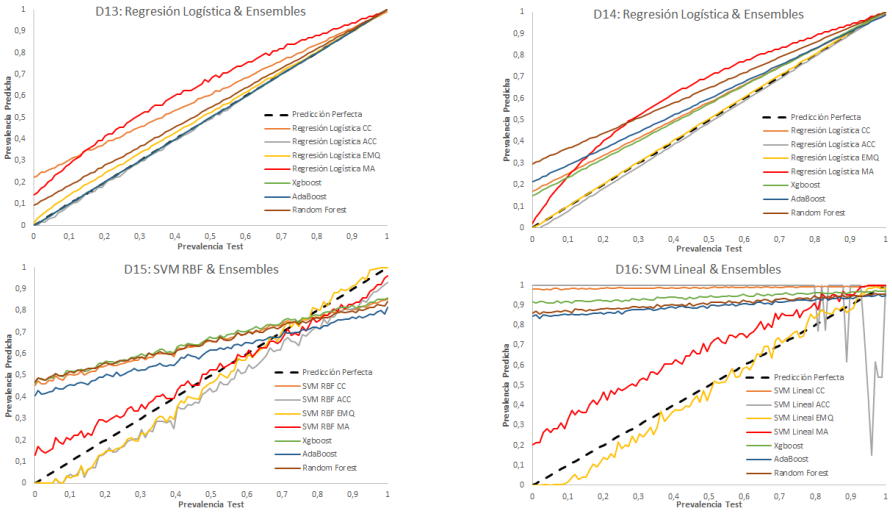

Figura 41: MAE Promedio Regresión Logística.

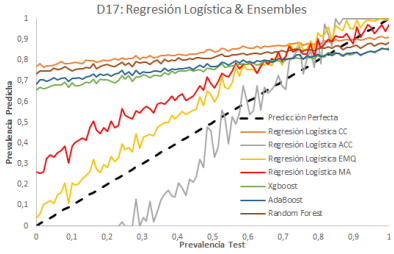

Figura 42: MAE Promedio Regresión Logística.

Tal como se puede observar en la mayor parte de los datasets se observa un mismo patrón de evolución a lo largo de las diferentes prevalencias simuladas de los métodos *Ensemble* evaluados, así tanto los métodos CC como los *Ensembles* evidencian comportamientos paralelos en su evolución (esto se puede ver nitidamente en los conjunto de datos 7, 8, 9 o 15). Es decir, los métodos *Ensemble* aun siendo más precisos que los métodos CC se comportan de la misma manera que éstos, es decir, suelen sobrestimar o infraestimar en regiones de prevalencia similares a las que lo hacen los métodos CC. Esto implica que en aquellos conjuntos de datos en los que los métodos CC fallan a la hora de predecir las prevalencias, o lo que es equivalente no son paralelos a la recta de predicción perfecta, los métodos *Ensemble* también cometeran el mismo error, y por tanto será necesario practicar algún tipo de ajuste en las predicciones para paliar esos errores.

4.2.7. Ajuste CCA en Xgboost

Dadas las debilidades apuntadas de los métodos *Ensemble* en la comparación con las técnicas de *Cuantificación* más avanzadas, hemos incorporado un análisis adicional que consiste en practicar un ajuste análogo al CCA[28] para el XGboost y analizar si esta corrección mejora sensiblemente el comportamiento del algoritmo. Los resultados obtenidos son plenamente satisfactorios como veremos a continuación.
En la comparativa con el XGboost sale victoriosa la versión de éste con el ajuste CCA:

Conjunto	Ensemble	
	Xgboost	Xgboost CCA
D01	0,0384	0,0288
D02	-	-
D03	0,0321	0,0566
D04	0,0488	0,0577
D05	0,0230	0,0118
D06	0,1048	0,0301
D07	0,1498	0,0224
D08	0,2210	0,0516
D09	0,1998	0,0722
D10	-	-
D11	0,0179	0,0033
D12	0,0364	0,0097
D13	0,0009	0,0049
D14	0,0745	0,0158
D15	0,2011	0,1050
D16	0,4420	0,4464
D17	0,2877	0,4999

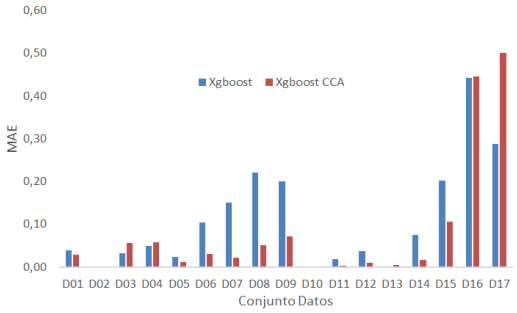

Figura 43: Comparativa Xgboost vs XGboost CCA.

En la comparativa se puede observar en la figura 43 que el ajuste practicado permite obetener mejores resultados en 12 de los 17 conjuntos con reducciones significativas en los valores del MAE alcanzando un promedio del 65 % sobre los conjuntos en los que el XGboost CCA bate al XGboost. Este mejoría se traslada a la comparativa con los métodos tradicionales tal como podemos observar en las figuras 44, 45 y 46:

[28]Se aplicara la misma metodología que la descrita en la Metodología Experimental para esta técnica.

Conjunto	Cuantificadores (SVM Lineal)				Ensembles			Ensembles Modificado
	CC	CCA	EMQ	MA	Xgboost	Random Forest	Ada Boost	Xgboost CCA
D01	0,0218	0,0191	0,0202	0,0494	0,0384	0,0119	0,0181	0,0288
D02	-	-	0,0523	0,0467	-	-	-	-
D03	0,2577	0,0916	0,1259	0,0640	0,0321	0,0321	0,0321	0,0566
D04	0,0485	0,0420	0,0698	0,0208	0,0488	0,0642	0,0488	0,0577
D05	0,0483	0,0100	0,0116	0,0084	0,0230	0,1137	0,0329	0,0118
D06	0,1353	0,1107	0,0168	0,0439	0,1048	0,1043	0,1096	0,0301
D07	0,2743	0,0768	0,0266	0,0420	0,1498	0,2272	0,1727	0,0224
D08	0,1782	0,4298	0,1181	0,0977	0,2210	0,3844	0,2169	0,0516
D09	0,2994	0,0432	0,0826	0,1275	0,1998	0,2979	0,2238	0,0722
D10	0,0196	0,0065	0,0110	0,0033	-	0,0147	0,0196	-
D11	0,0746	0,0322	0,0069	0,0517	0,0179	0,0426	0,0241	0,0033
D12	0,1340	0,0777	0,0226	0,0867	0,0364	0,0481	0,0412	0,0097
D13	0,1188	0,0136	0,0122	0,1898	0,0009	0,0464	0,0009	0,0049
D14	0,1148	0,0214	0,0284	0,1281	0,0745	0,1489	0,1004	0,0158
D15	0,2234	0,0801	0,0546	0,1145	0,2011	0,1999	0,1726	0,1050
D16	0,4871	0,5371	0,0402	0,1660	0,4420	0,4111	0,3981	0,4464
D17	0,4615	0,5001	0,1059	0,1461	0,2877	0,3318	0,3019	0,4999
Ranking Promedio	**2,235**	**4,353**	**5,588**	**4,647**	**4,529**	**3,353**	**4,294**	**5,706**

Error Mínimo

Figura 44: MAE Promedio SVM Kernel Lineal con Xgboost CCA.

Conjunto	Cuantificadores (SVM RBF)				Ensembles			Ensembles Modificado
	CC	CCA	EMQ	MA	Xgboost	Random Forest	Ada Boost	Xgboost CCA
D01	0,0232	0,0067	0,0169	0,0446	0,0384	0,0119	0,0181	0,0288
D02	-	-	0,0503	0,0193	-	-	-	-
D03	0,0343	0,0477	0,0833	0,0340	0,0321	0,0321	0,0321	0,0566
D04	0,0495	0,0485	0,0557	0,1309	0,0488	0,0642	0,0488	0,0577
D05	0,0500	0,0075	0,0144	0,0105	0,0230	0,1137	0,0329	0,0118
D06	0,1209	0,0332	0,0308	0,0616	0,1048	0,1043	0,1096	0,0301
D07	0,1954	0,0499	0,0277	0,0311	0,1498	0,2272	0,1727	0,0224
D08	0,2977	0,1471	0,1262	0,0659	0,2210	0,3844	0,2169	0,0516
D09	0,2589	0,0611	0,0351	0,0410	0,1998	0,2979	0,2238	0,0722
D10	0,0347	0,0043	0,0110	0,0063	-	0,0147	0,0196	-
D11	0,0415	0,0093	0,0224	0,0343	0,0179	0,0426	0,0241	0,0033
D12	0,0799	0,0232	0,0209	0,1193	0,0364	0,0481	0,0412	0,0097
D13	0,1085	0,0337	0,0398	0,1871	0,0009	0,0464	0,0009	0,0049
D14	0,0925	0,0254	0,0318	0,1382	0,0745	0,1489	0,1004	0,0158
D15	0,1934	0,0722	0,0397	0,0536	0,2011	0,1999	0,1726	0,1050
D16	0,4175	0,1350	0,0590	0,2840	0,4420	0,4111	0,3981	0,4464
D17	0,3027	0,2363	0,1062	0,2509	0,2877	0,3318	0,3019	0,4999
Ranking Promedio	**2,647**	**6,059**	**5,588**	**4,235**	**4,294**	**2,647**	**4,000**	**5,294**

Error Mínimo

Figura 45: MAE Promedio SVM Kernel RBF con Xgboost CCA.

Conjunto	Cuantificadores (Regresión Logística)				Ensembles			Ensembles Modificado
	CC	CCA	EMQ	MA	Xgboost	Random Forest	Ada Boost	Xgboost CCA
D01	0,0521	0,0391	0,0194	0,0533	0,0384	0,0119	0,0181	0,0288
D02	-	-	0,0114	0,0206	-	-	-	-
D03	0,3210	0,1930	0,1043	0,1914	0,0321	0,0321	0,0321	0,0566
D04	0,0607	0,0308	0,0338	0,2552	0,0488	0,0642	0,0488	0,0577
D05	0,0421	0,0118	0,0052	0,0186	0,0230	0,1137	0,0329	0,0118
D06	0,1193	0,0604	0,0114	0,0573	0,1048	0,1043	0,1096	0,0301
D07	0,2428	0,0638	0,0457	0,0409	0,1498	0,2272	0,1727	0,0224
D08	0,3808	0,0529	0,1640	0,0728	0,2210	0,3844	0,2169	0,0516
D09	0,2601	0,1246	0,0958	0,1164	0,1998	0,2979	0,2238	0,0722
D10	0,0196	0,0065	0,0667	0,0027	-	0,0147	0,0196	-
D11	0,0722	0,0165	0,0096	0,0414	0,0179	0,0426	0,0241	0,0033
D12	0,1220	0,0523	0,0143	0,1329	0,0364	0,0481	0,0412	0,0097
D13	0,1086	0,0058	0,0212	0,1432	0,0009	0,0464	0,0009	0,0049
D14	0,0819	0,0127	0,0045	0,1432	0,0745	0,1489	0,1004	0,0158
D15	0,1889	0,1274	0,0956	0,1366	0,2011	0,1999	0,1726	0,1050
D16	0,4337	0,0511	0,0612	0,1780	0,4420	0,4111	0,3981	0,4464
D17	0,3524	0,1175	0,1056	0,1767	0,2877	0,3318	0,3019	0,4999
Ranking Promedio	**2,118**	**5,412**	**6,059**	**3,882**	**4,353**	**3,000**	**4,118**	**5,765**

Error Mínimo

Figura 46: MAE Promedio Regresión Logística con Xgboost CCA.

Tal como se puede observar en el ranking la mejora de aplicar el ajuste es significativa siendo el mejor algoritmo de *Cuantificación* cuando lo comparamos con un clasificador SVM con kernel lineal y el segundo mejor para el resto de algoritmos, esto es consecuencia de que en aproximadamente el 50 % de los conjuntos de datos obtiene el menor MAE para todos los clasificadores examinados.

5. Conclusiones y próximos pasos

La *Cuantificación* es un problema del aprendizaje automático que persigue la determinación de la distribución de probabilidad de un conjunto de clases en un conjunto sin etiquetar. A pesar de su potencial similaridad con la *Clasificación* existen diferencias en sus planteamientos: variación de la distribución subyacente, diferente función de pérdida...que devienen en el establecimiento de métodos propios, agregativos, que tomando como punto de partida un algoritmo de clasificación ajustan la salida para obtener una estimación más precisa de la distribución. La mayor parte de estudios en este punto se han llevado a cabo con clasificadores tracionales (SVM, regresión logística...) siendo de poca relevancia, al menos bibliográfica, la utilización de los métodos *Ensemble* a los problemas de *Clasificación*; se trata por tanto de un área de poca evolución hasta el momento, a pesar de las potenciales sinergias que existen entre ambos: si la *Cuantificación* tiene como una de sus características principales la variación de la distribución subyacente, la posibilidad de disponer de un amplio abanico de modelos base que han sido entrenados bajo diversas circunstancias, tal como puede realizar un algoritmo *Ensemble*, puede ser una buena herramienta para afrontar la *Cuantificación*. A pesar de esta complementariedad la literatura consultada sobre la aplicación de los *Ensembles* es reducida y siempre concentrada en un tipo específico de *Ensembles* (Random Forest, Baggins...), dejando de lado alternativas como los Boosting que pudieran ser de utilidad. El estudio llevado a cabo en este trabajo aborda esta área inexplorada hasta el momento: la aplicación de algoritmos tipo Boosting a los problemas de *Cuantificación*.

Los resultados de los análisis llevados a cabo muestran que los métodos *Ensemble* mejoran sensiblemente los resultados de los métodos *Classify and Count* (CC) sobre clasificadores tradicionales aplicados a la *Cuantificación*; obteniendo los mejores resultados los algoritmos de tipo Boosting, en concreto el Xgboost. Sin embargo los *Ensemble* muestran un comportamiento similar a los métodos CC en los diferentes niveles de prevalencia simulada, es decir, si los métodos CC sobrestiman un nivel de prevalencia los métodos *Ensemble* incurren en el mismo sesgo y vicecersa. No son capaces de revertir el desajuste entre lo observado y lo predicho; no obstante, el nivel de error es menor tal como mencionamos con anterioridad. Con el fin de paliar este defecto se ha procedido a practicar un ajuste basado en la matriz de confusión, *Classify and Count Adjusted* (CCA), idéntico al realizado en *Cuantificación*, al algoritmo *Ensemble*. La incorporación de este ajuste incrementa la efectividad del algoritmo Xgboost pasando a estar entre los dos algoritmos más precisos en los conjuntos de datos valorados.

La verificación de la bondad de los *Ensemble*, en concreto los Boosting, abre nuevas vías de investigación que permitiran profundizar en las conclusiones obtenidas en este documento. Dos son las vías que consideramos deberían ser exploradas en estudios posteriores:

1. Ampliar el estudio aquí realizado a más familias de algoritmos *Ensemble*

para poder generalizar las conclusiones aquí mostradas incorporando un fundamento teórico que permita explicar estos resultados. En esta misma línea sería de interés confrontar la metodología desarrollada por Peréz-Gallego, Quevedo y del Coz en [PQC17] con estas familias alternativas de *Ensemble*.

2. En el caso concreto del modelo Xgboost, y dada su buen comportamiento, consideramos que otra vía de mejorar su aplicabilidad sobre problemas de *Cuantificación* pasa por adaptar la función de pérdida del algoritmo a una propia de problemas de *Cuantificación*, es decir, implementar un cuantificador Xgboost.

6. Anexos

6.1. Demostración Proposición

Proposition *Sea $\mathcal{S} \subseteq \mathcal{X} \times \mathcal{Y}$ el conjunto de entrenamiento de un problema de clasificación binario en el que las ratio de verdaderos positivos (TPR) y falsos positivos (FPR) son constantes y sea p la frecuencia de casos positivos en esta muestra. Entonces si h^C es un clasificador imperfecto, $TPR \neq FPR$, entrenado en este conjunto que verifica:*

$$h^C(\mathcal{S}) = p$$

ante cualquier variación de la probabilidad de positivos, p', en el conjunto test, $\mathcal{T}$, el clasificador no se mantendrá calibrado:

$$h^C(\mathcal{T}) \neq p'$$

Demostración. De cara a la demostración adaptaremos la notación en términos de probabilidad, así:

$$P_{\mathcal{T}}(\{h = 1\}) := h^C(\mathcal{T})$$

Entonces:

$$\begin{aligned} &P_{\mathcal{T}}(\{h = 1\}) = P_{\mathcal{T}}(\{h = 1\}|1)P_{\mathcal{T}}(1) + P_{\mathcal{T}}(\{h = 1\}|0)P_{\mathcal{T}}(0) = \\ &= P_{\mathcal{T}}(\{h = 1\}|1)P_{\mathcal{T}}(1) + P_{\mathcal{T}}(\{h = 1\}|0)(1 - P_{\mathcal{T}}(1)) \implies \\ &P_{\mathcal{T}}(\{h = 1\}) = TPR \times P_{\mathcal{T}}(1) + FPR \times (1 - P_{\mathcal{T}}(1)) \end{aligned}$$

Teniendo en cuenta que el clasificador está calibrado en la muestra de entrenamiento:

$$P_{\mathcal{S}}(\{h = 1\}) = TPR \times P_{\mathcal{S}}(1) + FPR \times (1 - P_{\mathcal{S}}(1)) = P_{\mathcal{S}}(1)$$

y por tanto,

$$FPR + (TPR - FPR)P_{\mathcal{S}}(1) = P_{\mathcal{S}}(1) \implies P_{\mathcal{S}}(1) = \frac{FPR}{1 - TPR + FPR}$$

Es decir, para que el clasificador esté calibrado es necesario que se verifique esta condición. Teniendo en cuenta que por hipótesis tanto TPR como FPR son constantes la única probabilidad p en la que el clasificador estará calibrado será en $P_{\mathcal{S}}(1)$ y por tanto para cualquier $P_{\mathcal{T}}(1) \neq P_{\mathcal{S}}(1)$ el clasificador no seguirá calibrado.

6.2. Ranking

La elaboración del ranking se ha realizado de acuerdo al siguiente procedimiento:

1. Sean $\{r_i^j\}_{i,j=1}^{n,M}$ las medidas de error asociadas a n métodos sobre M conjuntos de datos.

2. Para cada conjunto de datos se ordena en sentido descendente las medidas de error:

$$r_{i_1}^j \geq r_{i_2}^j \geq ... \geq r_{i_n}^j$$

3. Para cada una de las medidas se le hace corresponder un natural, n_i, comenzando en 1, con las siguientes reglas:

   - Si i=1 entonces $r_1^j \longrightarrow n_1 = 1$
   - Si $r_{i-1}^j > r_i^j > r_{i+1}^j \longrightarrow n_{i+1}^j = n_i^j + 1$
   - Si $r_{i-1}^j \geq r_i^j = r_{i+1}^j \longrightarrow n_{i+1}^j = n_i^j$
   - Si $r_{i-1}^j = r_i^j > r_{i+1}^j \longrightarrow n_{i+1}^j = n_i^j + k_i$ siendo k_i el número de empates con r_i^j.

4. El ranking del método i se obtiene promediando sobre los M conjuntos de datos:

$$R(i) = \frac{1}{M} \sum_{j=1}^{M} n_i^j$$

7. Bibliografía

Referencias

[HS90] L.K. Hansen y P. Salamon. "Neural network ensembles". En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.10 (1990), págs. 993-1001. DOI: 10.1109/34.58871.

[FS97] Yoav Freund y Robert E Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". En: *Journal of Computer and System Sciences* 55.1 (1997), págs. 119-139. ISSN: 0022-0000. DOI: https://doi.org/10.1006/jcss.1997.1504. URL: https://www.sciencedirect.com/science/article/pii/S002200009791504X.

[Die00] Thomas G. Dietterich. "Ensemble methods in Machine Learning". En: *In Proceedings of the 1st International Workshop on Multiple Classifier Systems.* (2000), págs. 1-15.

[SLD02] Marco Saerens, Patrice Latinne y Christine Decaestecker. "Adjusting the Outputs of a Classifier to New a Priori Probabilities: A Simple Procedure". En: *Neural Comput.* 14.1 (ene. de 2002), págs. 21-41. ISSN: 0899-7667. DOI: 10.1162/089976602753284446. URL: http://dx.doi.org/10.1162/089976602753284446.

[For05] George Forman. "Counting Positives Accurately Despite Inaccurate Classification". En: *ECML*. 2005.

[For06] George Forman. "Quantifying Trends Accurately Despite Classifier Error and Class Imbalance". En: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '06. Philadelphia, PA, USA: ACM, 2006, págs. 157-166. ISBN: 1-59593-339-5. DOI: 10.1145/1150402.1150423. URL: http://doi.acm.org/10.1145/1150402.1150423.

[For07] George Forman. "Quantifying counts, costs and trends accurately via machine learning." En: Palo Alto, CA, USA, 2007.

[KL08] Gary King y Ying Lu. "Verbal Autopsy Methods with Multiple Causes of Death". En: *Statistical Science* 23 (2008), 78-91.

[XW09] Jack Chongjie Xue y Gary M. Weiss. "Quantification and Semi-supervised Classification Methods for Handling Changes in Class Distribution". En: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '09. Paris, France: ACM, 2009, págs. 897-906. ISBN: 978-1-60558-495-9. DOI: 10.1145/1557019.1557117. URL: http://doi.acm.org/10.1145/1557019.1557117.

[Bel+10] A. Bella y col. "Quantification via Probability Estimators". En: *2010 IEEE International Conference on Data Mining*. 2010, págs. 737-742. DOI: 10.1109/ICDM.2010.75.

[HK10] Daniel Hopkins y Gary King. “A Method of Automated Nonparametric Content Analysis for Social Science”. En: *American Journal of Political Science* 54.1 (ene. de 2010), 229-247.

[Gol+11] Edward Goldstein y col. “Estimating Incidence Curves of Several Infections Using Symptom Surveillance Data”. En: *PLOS ONE* 6.8 (ago. de 2011), págs. 1-8. DOI: 10.1371/journal.pone.0023380. URL: https://doi.org/10.1371/journal.pone.0023380.

[AML12] Yaser S. Abu-Mostafa, Malik Magdon-Ismail y Hsuan-Tien Lin. *Learning From Data*. AMLBook, 2012.

[Bar+12] Jose Barranquero y col. “On the study of nearest neighbor algorithms for prevalence estimation in binary problems”. En: *Pattern Recognition* 46 (ago. de 2012). DOI: 10.1016/j.patcog.2012.07.022.

[RV12] Matteo Re y Giorgio Valentini. “Ensemble methods: A review”. En: ene. de 2012, págs. 563-594.

[Zho12] Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. 1st. Chapman Hall/CRC, 2012. ISBN: 1439830037.

[Bel+13] Antonio Bella y col. “Aggregative quantification for regression”. En: *Data Mining and Knowledge Discovery* 28 (2013), págs. 475-518.

[GAA13] VıCtor GonzáLez-Castro, RocıO Alaiz-RodrıGuez y Enrique Alegre. “Class Distribution Estimation Based on the Hellinger Distance”. En: *Inf. Sci.* 218 (ene. de 2013), págs. 146-164. ISSN: 0020-0255. DOI: 10.1016/j.ins.2012.05.028. URL: https://doi.org/10.1016/j.ins.2012.05.028.

[Mil+13] L. Milli y col. “Quantification Trees”. En: *2013 IEEE 13th International Conference on Data Mining*. Dic. de 2013, págs. 528-536. DOI: 10.1109/ICDM.2013.122.

[BDC15] Jose Barranquero, Jorge Díez y Juan del Coz. “Quantification-oriented learning based on reliable classifiers”. En: *Pattern Recognition* 48 (feb. de 2015). DOI: 10.1016/j.patcog.2014.07.032.

[GS15] W. Gao y F. Sebastiani. “Tweet sentiment: From classification to quantification”. En: *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. Ago. de 2015, págs. 97-104. DOI: 10.1145/2808797.2809327.

[Mil+15] L. Milli y col. “Quantification in social networks”. En: *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. Oct. de 2015, págs. 1-10. DOI: 10.1109/DSAA.2015.7344845.

[DGS16] Giovanni Da San Martino, Wei Gao y Fabrizio Sebastiani. "QCRI at SemEval-2016 Task 4: Probabilistic Methods for Binary and Ordinal Quantification". En: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, jun. de 2016, págs. 58-63. DOI: 10.18653/v1/S16-1006. URL: https://aclanthology.org/S16-1006.

[Esu16] Andrea Esuli. "ISTI-CNR at SemEval-2016 Task 4: Quantification on an Ordinal Scale". En: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, jun. de 2016, págs. 92-95. DOI: 10.18653/v1/S16-1011. URL: https://aclanthology.org/S16-1011.

[Tas16] Dirk Tasche. "Does quantification without adjustments work?" En: (2016). DOI: 10.48550/ARXIV.1602.08780. URL: https://arxiv.org/abs/1602.08780.

[Vil+16] David Vilares y col. "LyS at SemEval-2016 Task 4: Exploiting Neural Activation Values for Twitter Sentiment Classification and Quantification". En: **SEMEVAL*. 2016.

[Gon+17] Pablo González y col. "A Review on Quantification Learning". En: *ACM Computing Surveys* 50 (sep. de 2017), págs. 1-40. DOI: 10.1145/3117807.

[PQC17] Pablo Pérez-Gállego, J. R. Quevedo y J. J. Coz. "Using ensembles for problems with characterizable changes in data distribution: A case study on quantification". En: *Inf. Fusion* 34 (2017), págs. 87-100.

[EFS18] Andrea Esuli, Alejandro Moreo Fernández y Fabrizio Sebastiani. "A Recurrent Neural Network for Sentiment Quantification". En: *CoRR* abs/1809.00836 (2018). arXiv: 1809.00836. URL: http://arxiv.org/abs/1809.00836.

[KLV18] Nikolay Karpov, Alexander Lyashuk y Arsenii Vizgunov. "Sentiment Analysis Using Deep Learning: NET 2017, Nizhny Novgorod, Russia, June 2017". En: ago. de 2018, págs. 281-288. ISBN: 978-3-319-96246-7. DOI: 10.1007/978-3-319-96247-4_20.

[Pér+18] Pablo Pérez-Gállego y col. "Dynamic Ensemble Selection for Quantification Tasks". En: *Information Fusion* 45 (ene. de 2018). DOI: 10.1016/j.inffus.2018.01.001.

[Seb18a] Fabrizio Sebastiani. "Evaluation Measures for Quantification: An Axiomatic Approach". En: *CoRR* abs/1809.01991 (2018). arXiv: 1809.01991. URL: http://arxiv.org/abs/1809.01991.

[Seb18b] Fabrizio Sebastiani. "Quantification, using supervised learning to estimate class prevalence." En: (2018).

[Seb18c] Fabrizio Sebastiani. "Sentiment Quantification of User-Generated Content". En: *Encyclopedia of Social Network Analysis and Mining*. Ed. por Reda Alhajj y Jon Rokne. New York, NY: Springer New York, 2018, págs. 2454-2465. ISBN: 978-1-4939-7131-2. DOI: 10.1007/978-1-4939-7131-2_110170. URL: https://doi.org/10.1007/978-1-4939-7131-2_110170.

[Mal+19] André Maletzke y col. "DyS: a Framework for Mixture Models in Quantification". En: mar. de 2019.

[MS20] Alejandro Moreo y Fabrizio Sebastiani. "Re-Assessing the Çlassify and Count"Quantification Method". En: *CoRR* abs/2011.02552 (2020). arXiv: 2011.02552. URL: https://arxiv.org/abs/2011.02552.

[MES21] Alejandro Moreo, Andrea Esuli y Fabrizio Sebastiani. "QuaPy: a python-based framework for quantification". En: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2021, págs. 4534-4543.

[Qi+21] Lei Qi y col. "A Framework for Deep Quantification Learning". En: *Machine Learning and Knowledge Discovery in Databases*. Ed. por Frank Hutter y col. Cham: Springer International Publishing, 2021, págs. 232-248. ISBN: 978-3-030-67658-2.

[Sak21] Tetsuya Sakai. "Evaluating Evaluation Measures for Ordinal Classification and Ordinal Quantification". En: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, ago. de 2021, págs. 2759-2769. DOI: 10.18653/v1/2021.acl-long.214. URL: https://aclanthology.org/2021.acl-long.214.

[Ayy+22] Kashif Ayyub y col. "A Feature-Based Approach for Sentiment Quantification Using Machine Learning". En: *Electronics* 11.6 (2022). ISSN: 2079-9292. DOI: 10.3390/electronics11060846. URL: https://www.mdpi.com/2079-9292/11/6/846.

[ABB] Giambattista Amati, Marco Bianchi y Fondazione Ugo Bordoni. *Sentiment Estimation on Twitter*.