Una mejora de los modelos apilados para la clasificación multi-etiqueta

Autores no revelados para garantizar una revisión ciega

Información de contacto oculta

Resumen El objetivo de la clasificación multi-etiqueta es obtener modelos capaces de asignar las etiquetas, de entre un conjunto predefinido, que mejor describen un nuevo objeto del dominio. El enfoque más directo consiste en construir un clasificador independiente para cada etiqueta; es el método conocido como relevancia binaria. Sin embargo, estudios previos demuestran que, para obtener modelos más precisos, durante el proceso inductivo se deben tener en cuenta las posibles dependencias que existen entre las etiquetas. Uno de los métodos propuestos para este fin se basa en el *apilamiento* de modelos, de forma que se intentan detectar las correlaciones entre etiquetas en el último nivel de la pila, formado por los clasificadores que producen la predicción final. Este artículo estudia este método, analizando sus propiedades y comparándolo con otras aproximaciones más recientes. Además, se propone una modificación que permite mejorar su capacidad predictiva para alguna de las medidas más importantes empleadas en la clasificación multi-etiqueta.

1. Introduction

En la actualidad toda la información multimedia, desde los textos, a los vídeos, pasando por la música o las películas, se etiqueta para facilitar a los usuarios la búsqueda de los contenidos que desean. Las etiquetas asignadas pertenecen a conjuntos previamente definidos y su utilidad es dar una descripción rápida del contenido del elemento. Dentro del campo del aprendizaje, la tarea de producir modelos capaces de asignar automáticamente estos subconjuntos de etiquetas a nuevos objetos recibe el nombre de clasificación multi-etiqueta. Desde un punto de vista formal, su principal diferencia con respecto a la clasificación tradicional es que los objetos pueden pertenecer simultáneamente a más de una clase, lo que impide que se puedan aplicar directamente los clasificadores multi-clase conocidos. Por ese motivo diversas técnicas de aprendizaje han sido adaptadas para tratar este nuevo problema, como los árboles de decisión [3], los algoritmos basados en instancias [15] o las máquinas de vectores soporte [5].

El método denominado relevancia binaria (BR, *binary relevance*) es el más simple para abordar este problema de aprendizaje y suele ser empleado como algoritmo base para comparar el rendimiento de los nuevos métodos desarrollados. Consiste en construir un clasificador independiente para cada etiqueta. La principal crítica al algoritmo BR es precisamente la asunción que realiza al considerar que las etiquetas son independientes entre sí, lo cual no es cierto en

muchas aplicaciones reales. En los conjuntos de datos disponibles se ha observado que existen dependencias entre las etiquetas y que para mejorar el rendimiento de los clasificadores es necesario tenerlas en cuenta.

Por este motivo, gran parte de la investigación actual en el campo de la clasificación multi-etiqueta se centra en desarrollar nuevos algoritmos que detecten y exploten dichas dependencias. Los métodos propuestos podemos clasificarlos atendiendo a dos criterios: i) el tamaño de los subconjuntos de etiquetas entre las que se buscan dependencias, y ii) el tipo de correlaciones que se tratan de encontrar. De acuerdo con el primer aspecto, nos encontramos con algoritmos que consideran las relaciones entre pares de etiquetas [5,6], y otros que buscan correlaciones entre subconjuntos más amplios [10,13], incluyendo los que estudian la influencia del resto de etiquetas al tratar de predecir una de ellas [2,8]. Por otro lado, de acuerdo con el tipo de relaciones buscadas [4], hay algoritmos diseñados para detectar la dependencia condicional (aquella que se da para una instancia concreta), por ejemplo [4,10,12]; y la dependencia incondicional (un tipo de dependencia global, que no depende de observaciones individuales) [2,8].

En este trabajo se profundiza en uno de los métodos capaces de inducir modelos que tienen en cuenta la dependencia entre etiquetas. Se trata del algoritmo propuesto por Godbole y Sharawagi [8] cuya principal aportación consiste en emplear modelos *apilados* [14], técnica conocida como *stacking*, en el contexto de la clasificación multi-etiqueta. Este paradigma de aprendizaje se basa en construir modelos compuestos por grupos apilados de clasificadores, de forma que las salidas de los clasificadores de un nivel son usadas como entradas en los del nivel siguiente en la pila. La predicción final la determinan los clasificadores de la cima de la pila. La idea de los autores es realizar modelos con dos niveles, en el primero se obtienen clasificadores independientes que predicen cada etiqueta, iguales que los del BR, y sus salidas son empleadas en el segundo nivel para construir clasificadores más complejos que tratan de capturar las dependencias entre etiquetas. Este artículo demuestra que este enfoque no permite descubrir todas las correlaciones que existen entre las etiquetas y que la idea del apilamiento introduce errores acumulativos que degradan su rendimiento. Propondremos un nuevo método que incrementa el nivel de acierto para alguna de las medidas empleadas en la clasificación multi-etiqueta.

El resto del artículo está organizado como sigue. La siguiente sección introduce formalmente la clasificación multi-etiqueta. La sección 3 describe los métodos basados en modelos apilados, incluyendo nuestra propuesta. El artículo se cierra con un estudio experimental y algunas conclusiones.

2. Clasificación Multi-etiqueta

Dado un conjunto no vacío de etiquetas $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_m\}$, y siendo $\mathcal{X}$ un cierto espacio de entrada, una tarea de clasificación multi-etiqueta viene dada por un conjunto de entrenamiento $S = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$, cuyos ejemplos fueron obtenidos de forma independiente y aleatoria de una distribución de probabilidad desconocida $\mathbf{P}(\mathbf{X}, \mathbf{Y})$ en $\mathcal{X} \times \mathcal{Y}$, y en donde el espacio de salida $\mathcal{Y}$ es el

conjunto de partes de $\mathcal{L}$, $\mathcal{P}(\mathcal{L})$. Para hacer la notación más sencilla de entender, definimos $\boldsymbol{y}_i = \{y_1, y_2, \ldots, y_m\}$ como un vector binario, donde cada componente $y_j = 1$ indica la presencia de la etiqueta ℓ_j en el conjunto de etiquetas *relevantes* para $\boldsymbol{x}_i$. Usando esta convención, el espacio de salida puede definirse también como $\mathcal{Y} = \{0,1\}^m$.

El objetivo de la clasificación multi-etiqueta es obtener a partir de S una hipótesis $\boldsymbol{h} : \mathcal{X} \longrightarrow \mathcal{Y}$, que asigne el subconjunto de etiquetas correcto para una nueva instancia $\boldsymbol{x}$. Algunos métodos se basan en descomponer $\boldsymbol{h}$ en un conjunto de sub-hipótesis, $\boldsymbol{h}(\boldsymbol{x}) = (h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \ldots h_m(\boldsymbol{x}))$, una por cada etiqueta, donde cada h_j está definida como:

$$h_j : \mathcal{X} \longrightarrow \{0,1\}, \tag{1}$$

y es capaz de predecir si la etiqueta ℓ_j debe ser asignada o no al ejemplo $\boldsymbol{x}$. Este es el caso del algoritmo BR que aprende un clasificador binario h_j por cada etiqueta en el que solamente se emplea la información de esa etiqueta ignorando el resto. A pesar de su simplicidad, el método BR presenta varias ventajas: i) pueden emplearse muchos métodos para obtener los clasificador binarios h_j, ii) tiene una complejidad lineal con respecto al número de etiquetas, y iii) es fácilmente paralelizable. Su desventaja es obvia, no considera ninguna dependencia que pueda existir entre las etiquetas, lo que degrada su rendimiento cuando dichas correlaciones ocurren. A pesar de todo, cuando se usan buenos algoritmos para inducir los clasificadores binarios h_j, con algún proceso para fijar sus parámetros, BR obtiene resultados aceptables, e incluso para algunas medidas resultados muy competitivos, como es el caso de la función de pérdida *Hamming* (Eq. 6).

Para medir el rendimiento de los clasificadores multi-etiqueta se emplean una gran variedad de medidas. En [12] se puede encontrar una presentación de todas ellas, incluyendo su categorización por varios criterios: basadas en ejemplos, basadas en etiquetas y basadas en el orden de las etiquetas. En este trabajo solamente consideraremos medidas basadas en ejemplos, ya que, por una parte, algunas fueron las propuestas por los autores de los modelos basados en apilamiento [8], y por otro lado, nos permiten estudiar si los métodos comparados capturan o no las dependencias entre etiquetas. La mayoría de ellas tienen su origen en el campo de la recuperación de información[1]:

- **Acierto** (*accuracy*), calcula el porcentaje de etiquetas relevantes predichas en el subconjunto formado por la unión de las etiquetas asignadas por el clasificador y las relevantes[2],

$$Acierto(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) = \frac{\sum_{i=1}^{m} [\![y_i = 1 \text{ and } h_i(\boldsymbol{x}) = 1]\!]}{\sum_{i=1}^{m} [\![y_i = 1 \text{ or } h_i(\boldsymbol{x}) = 1]\!]}. \tag{2}$$

- **Precisión** (*precision*), determina la fracción de etiquetas relevantes dentro de las etiquetas predichas,

$$Precision(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) = \frac{\sum_{i=1}^{m} [\![y_i = 1 \text{ and } h_i(\boldsymbol{x}) = 1]\!]}{\sum_{i=1}^{m} [\![h_i(\boldsymbol{x}) = 1]\!]}. \tag{3}$$

[1] Indicamos el nombre original en inglés.

[2] La expresión $[\![p]\!]$ es 1 si el predicado p es cierto, y 0 en otro caso.

- **Exhaustividad** (*recall*) es la proporción de etiquetas relevantes del ejemplo que son correctamente predichas,

$$Exhaustividad(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) = \frac{\sum_{i=1}^{m} [\![y_i = 1 \text{ and } h_i(\boldsymbol{x}) = 1]\!]}{\sum_{i=1}^{m} [\![y_i = 1]\!]}. \tag{4}$$

- $\boldsymbol{F_1}$ es la media armónica de Precisión y Exhaustividad,

$$F_1(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) = \frac{2 \sum_{i=1}^{m} [\![y_i = 1 \text{ and } h_i(\boldsymbol{x}) = 1]\!]}{\sum_{i=1}^{m} ([\![y_i = 1]\!] + [\![h_i(\boldsymbol{x}) = 1]\!])}. \tag{5}$$

Todas las métricas descritas anteriormente presentan un sesgo que favorece a los métodos que asignan correctamente las etiquetas relevantes. Además, el rendimiento de este tipo de clasificadores se suele medir mediante otras dos medidas:

- **Pérdida Hamming**, definida como la proporción de etiquetas cuya relevancia es incorrectamente predicha,

$$Hamming(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) = \frac{1}{m} \sum_{i=1}^{m} [\![y_i \neq h_i(\boldsymbol{x})]\!]. \tag{6}$$

- **Error 0/1**, determina si los subconjuntos de etiquetas relevantes y predichas son iguales o no,

$$Cero - Uno(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) = [\![\boldsymbol{y} \neq \boldsymbol{h}(\boldsymbol{x})]\!]. \tag{7}$$

3. Métodos basados en modelos apilados

Godbole y Sharawagi presentan un método [8] basado en utilizar modelos apilados [14], que trata de solventar el problema de la independencia entre las etiquetas del BR. Durante el entrenamiento, el método construye dos grupos de clasificadores apilados. El primero está formado por los mismos clasificadores binarios del BR, los denotaremos por $\boldsymbol{h}^1(\boldsymbol{x}) = (h_1^1(\boldsymbol{x}), \ldots, h_m^1(\boldsymbol{x}))$. En un segundo nivel, llamado meta-nivel, se construye otro grupo de clasificadores, de nuevo uno por etiqueta. Pero en este caso se emplea un espacio de atributos aumentado:

$$h_j^2 : \mathcal{X} \times \{0,1\}^m \longrightarrow \{0,1\}, \tag{8}$$

donde los m nuevos atributos se corresponden con las salidas de los clasificadores del primer nivel. Es decir, $\boldsymbol{h}^2(\boldsymbol{x}, \boldsymbol{y'}) = (h_1^2(\boldsymbol{x}, \boldsymbol{y'}), \ldots, h_m^2(\boldsymbol{x}, \boldsymbol{y'}))$, donde $\boldsymbol{y'} = \boldsymbol{h}^1(\boldsymbol{x})$. El objetivo es que estos segundos clasificadores capturen las relaciones que existen entre las etiquetas. A la hora de obtener la predicción de un nuevo ejemplo se devuelven las salidas de los clasificadores del meta-nivel, $\boldsymbol{h}^2(\boldsymbol{x}, \boldsymbol{h}^1(\boldsymbol{x}))$. Por tanto, las salidas de los clasificadores del primer nivel, $\boldsymbol{h}^1(\boldsymbol{x})$, se emplean solamente para obtener los valores de los atributos del espacio aumentado.

3.1. Mejora propuesta

Si analizamos los modelos basados en apilamiento desde un punto de vista probabilístico, los clasificadores binarios del meta-nivel estiman $\mathbf{P}(y_j|\boldsymbol{x}, \boldsymbol{y'})$, siendo $\boldsymbol{y'}$ a su vez una estimación del conjunto de etiquetas relevantes que depende del objeto $\boldsymbol{x}$. Esta cadena de estimaciones pueden explicar por qué quizás $\boldsymbol{y'}$ no contiene la información adecuada para poder inferir la dependencia de la etiqueta y_j con respecto al resto. Aunque los modelos apilados son absolutamente formales desde la perspectiva del aprendizaje, la fuente de datos es la misma tanto en el entrenamiento como en el test (las salidas del primer nivel), si pensamos en la fiabilidad de los datos de entrenamiento, los datos de las etiquetas contenidas en el conjunto de entrenamiento son menos ruidosos y contienen la información **real** acerca de las relaciones entre las etiquetas. Las salidas del primer nivel $\boldsymbol{y'}$ contendrán los errores propios de los modelos con que se obtienen, que a su vez pueden producir nuevos errores en los clasificadores del meta-nivel.

Por todo ello, el método que proponemos se basa en modificar la forma de aprender los clasificadores del meta-nivel. En lugar de usar las predicciones de los modelos del primer nivel, emplearemos la información real sobre las demás etiquetas, es decir, la contenida en el conjunto de entrenamiento, eliminando obviamente la información de la propia etiqueta j. Nuestros clasificadores h_j^2 tendrán la forma:

$$h_j^2 : \mathcal{X} \times \{0,1\}^{m-1} \longrightarrow \{0,1\}, \tag{9}$$

donde el espacio de atributos se completará con la información real de las $m-1$ etiquetas restantes. Es decir, el grupo de clasificadores será $\boldsymbol{h}^2(\boldsymbol{x}, \boldsymbol{y}) = (h_1^2(\boldsymbol{x}, y_2, \ldots, y_m), \ldots, h_m^2(\boldsymbol{x}, y_1, \ldots, y_{m-1}))$. De acuerdo con la clasificación descrita en la sección 1, estos clasificadores tratan de detectar las dependencias *condicionales* entre *todas* las etiquetas. En teoría, la estimación que realizan, $\mathbf{P}(y_j|\boldsymbol{x}, y_1, \ldots, y_{j-1}, y_{j+1}, \ldots, y_m)$, debe ser más precisa que la de los modelos apilados, $\mathbf{P}(y_j|\boldsymbol{x}, \boldsymbol{y'})$. Trataremos de demostrarlo experimentalmente (sección 4).

3.2. Otros métodos relacionados

Algunas variantes de [8] han sido propuestas, mayoritariamente centradas en reducir el tamaño del espacio de características empleado en el meta-nivel. El objetivo es ignorar la información de aquellas etiquetas que no están relacionadas con las etiqueta j del modelo h_j^2 que se está aprendiendo. Por ejemplo, en [11] los autores proponen calcular el coeficiente Chi entre cada par de etiquetas (j, k) usando los datos del conjunto de entrenamiento. El método elimina la información de todas las etiquetas k cuya correlación con la etiqueta j esté por debajo de un cierto umbral. Este método incrementa la eficiencia computacional sin una pérdida, ni tampoco una mejora, significativa de la capacidad de acierto.

En [1] los autores proponen un nuevo algoritmo que incluye dos modificaciones al método apilado original: i) eliminan la estimación del clasificador h_j^1 cuando se construye el modelo h_j^2, es decir, no incluyen información de la propia clase, y, ii) en la fase de predicción usan un orden predefinido para la estimación de cada etiqueta basada en su frecuencia de aparición.

Read et al. describen [10] los llamados Clasificadores Cadena (CC), que pueden modelar las correlaciones entre las etiquetas manteniendo una complejidad del mismo orden que la del BR. Durante el entrenamiento, CC ordena las etiquetas en una cadena y construye m clasificadores binarios, uno por cada etiqueta. El espacio de características de cada clasificador se aumenta con la información de las etiquetas de los eslabones anteriores en la cadena. Por ejemplo, si la cadena sigue el mismo orden que el conjunto de etiquetas, entonces el clasificador h_j será:

$$h_j : \mathcal{X} \times \{0,1\}^{j-1} \longrightarrow \{0,1\}, \tag{10}$$

empleando los datos de las etiquetas previas en la cadena, $y_1, \ldots, y_{j-1}$, para aumentar el espacio de atributos. Para realizar una predicción, los clasificadores se aplican siguiendo el orden de la cadena, de forma que las salidas de los clasificadores anteriores son usadas para aumentar los sucesivos espacios de características de los clasificadores siguientes. Continuando con el ejemplo, tendríamos que $\boldsymbol{h}(\boldsymbol{x}) = (h_1(\boldsymbol{x}), h_2(\boldsymbol{x}, h_1(\boldsymbol{x})), h_3(\boldsymbol{x}, h_1(\boldsymbol{x}), h_2(\boldsymbol{x}, h_1(\boldsymbol{x}))), \ldots)$. Obviamente, el orden de la cadena influye en el rendimiento del modelo final. Aunque podrían emplearse heurísticos para determinarlo, los autores proponen ensamblar varios clasificadores cadena (ECC) usando diferentes ordenaciones de las etiquetas y diferentes muestras del conjunto de datos.

Nótese que el método CC tiene en común con nuestra propuesta que emplea los datos de entrenamiento reales de las etiquetas para construir los espacios de atributos aumentados. La diferencia es que no usa todas las etiquetas, sino solamente las anteriores según el orden de la cadena. Esto puede producir estimaciones pobres, especialmente cuando las etiquetas de las que depende una cierta etiqueta no están colocadas antes en la cadena, situación que será más frecuente para los clasificadores/etiquetas de los primeros eslabones. De hecho, CC asume un dependencia "en cadena" entre las etiquetas, cuando en la realidad las relaciones entre ellas tenderán a ser mucho más complejas. En nuestro método está garantizado que al construir el clasificador de una etiqueta siempre se tiene la información de todas las etiquetas de las que depende, ya que se incluye la información de todas ellas, aunque por otro lado, detectar las dependencias puede resultar en ciertos casos más complejo al emplearse espacios de más dimensiones.

Existen otros métodos, menos relacionados con los anteriores, que también tratan de encontrar interdependencias entre las etiquetas. RAkEL(RAndom kLabELsets) [13] construye iterativamente y combina varios clasificadores LP (*Label Power-set*). Un clasificador LP considera cada subconjunto de etiquetas que se dan en el conjunto de entrenamiento como una de las clases individuales de un nuevo problema multi-clase. En cada iteración i, RAkEL selecciona aleatoriamente, sin reemplazamiento, un conjunto de etiquetas $\boldsymbol{Y}_i$ de tamaño k. Después, aprende un clasificador LP de la forma $\mathcal{X} \rightarrow \mathcal{P}(\boldsymbol{Y}_i)$. Finalmente, mediante un proceso de votación se determinan las etiquetas asignadas. IBLR (Instance-Based Learning by Logistic Regression) [2] unifica el aprendizaje basado en instancias y la regresión logística. Centrándonos en la forma de detectar las dependencias, la idea principal consiste en extender la descripción de cada ejemplo $\boldsymbol{x}$ añadiendo atributos que expresan la presencia de cada etiqueta en el vecindario de $\boldsymbol{x}$.

Tabla 1. Propiedades de los conjuntos de datos usados en los experimentos

Conjunto	Atributos	Ejemplos	Etiquetas	Cardinalidad
bibtex	1836	7395	159	2.40
emotions	72	593	6	1.87
enron	1001	1702	53	3.38
genbase	1185	662	27	1.25
image	135	2000	5	1.24
mediamill	120	5000	101	4.27
medical	1449	978	45	1.25
reuters	243	7119	7	1.24
scene	294	2407	6	1.07
slashdot	1079	3782	22	1.18
yeast	103	2417	14	4.24

4. Resultados experimentales

Los experimentos realizados se diseñaron con dos objetivos. En primer lugar, estudiar el comportamiento de nuestro método y de los modelos apilados originales, y en segundo lugar, realizar una comparación amplia con los algoritmos que constituyen el estado del arte en la clasificación multi-etiqueta. Se utilizaron varios conjuntos de datos de dominio público, cuyas características aparecen en la tabla 1. Como puede apreciarse, son bastante diferentes en cuanto a número de atributos, ejemplos, etiquetas y cardinalidad (nº de etiquetas por ejemplo).

Los algoritmos comparados fueron: el algoritmo BR, el método basado en modelos apilados original [8] (denotado como STA), nuestra propuesta (STA$^{\boldsymbol{y}}$), y los algoritmos del estado del arte: MLkNN [15], RAkEL [13], IBLR [2] y ECC [10], en la versión descrita en [4], denominada como ECC*. Incluimos ECC* en lugar del algoritmo CC ya que obtiene mejores resultados [4]. Se empleó regresión logística [9] como el algoritmo de aprendizaje base con el que se construyen los clasificadores binarios individuales en los métodos que lo requieren. En todos los casos, el parámetro de regularización C se estableció para cada clasificador binario de forma individual mediante una búsqueda en $C \in [10^{-3}, \ldots, 10^{3}]$ optimizando el acierto en clasificación binaria, que se estimó mediante una validación cruzada estratificada de 2 particiones repetida 5 veces.

La tabla 2 y la tabla 3 contienen los resultados de todos los sistemas con las medidas de error descritas en la sección 2. Los resultados, mostrados como porcentajes, corresponden a validaciones cruzadas estratificadas de 10 particiones. El orden de los sistemas se muestra entre paréntesis. En caso de empate se asigna el promedio. La media de las posiciones de cada sistema se muestra en la última fila de la tabla correspondiente a cada medida. Siguiendo las recomendaciones de [7] se realizó una comparación estadística en dos pasos. En primer lugar, se utilizó un test de Friedman para rechazar la hipótesis nula de que todos los métodos obtienen un rendimiento igual. En el segundo paso se realizaron com-

Tabla 2. Resultados de todos los sistemas para Precisión, Exhaustividad, F_1 y Acierto

Precisión	BR	STA	STA^{y}	ECC*	MLkNN	IBLR	RAkEL
Bibtex	48,19(1)	48,01(2)	47,17(4)	47,69(3)	26,60(7)	28,92(6)	47,08 (5)
Emotions	56,36(4)	56,58(3)	62,09(2)	56,26(5)	52,42(7)	67,54(1)	52,79 (6)
Enron	69,99(1)	65,18(4)	65,51(2)	65,19(3)	54,90(6)	52,75(7)	56,05 (5)
Genbase	99,52(3,5)	99,40(5)	99,60(1)	99,52(3,5)	97,70(7)	98,90(6)	99,57 (2)
Image	44,23(7)	44,54(5)	53,88(1)	45,99(4)	44,33(6)	48,52(2)	46,66 (3)
Mediamill	78,81(2)	43,49(7)	70,11(6)	77,87(3)	76,93(4)	73,52(5)	80,40 (1)
Medical	78,94(5)	79,47(4)	81,33(1)	81,04(2)	62,43(7)	63,40(6)	80,79 (3)
Reuters	85,79(5)	85,89(4)	87,50(2)	86,41(3)	82,23(6)	70,71(7)	89,62 (1)
Scene	61,46(7)	67,13(5)	66,14(6)	67,45(4)	69,71(2)	71,40(1)	69,69 (3)
Slashdot	46,06(4)	47,91(3)	53,20(1)	42,79(5)	6,15(7)	8,09(6)	50,91 (2)
Yeast	71,13(3)	70,81(4)	66,80(7)	70,58(5)	72,92(1)	71,75(2)	68,62 (6)
Prom. rank.	(3,86)	(4,18)	(3,00)	(3,68)	(5,45)	(4,45)	(3,36)
Exhaus.	BR	STA	STA^{y}	ECC*	MLkNN	IBLR	RAkEL
Bibtex	33,80(4)	35,49(2)	34,82(3)	33,78(5)	14,06(7)	21,50(6)	41,97 (1)
Emotions	48,16(6)	49,81(4)	65,84(1)	48,92(5)	37,73(7)	64,54(2)	57,19 (3)
Enron	50,50(4)	56,68(2)	57,48(1)	45,79(5)	37,04(7)	38,04(6)	54,07 (3)
Genbase	99,07(4,5)	99,07(4,5)	99,07(4,5)	99,07(4,5)	94,96(7)	99,14(2)	99,57 (1)
Image	43,32(6)	43,54(5)	53,68(1)	44,10(3)	39,11(7)	43,68(4)	49,73 (2)
Mediamill	52,33(5)	60,02(1)	54,34(3)	51,25(6)	53,78(4)	56,69(2)	49,57 (7)
Medical	78,34(5)	83,09(1)	81,02(2)	79,01(4)	59,01(7)	65,05(6)	81,00 (3)
Reuters	84,90(5)	84,93(4)	90,29(1)	85,19(3)	81,09(6)	69,45(7)	89,63 (2)
Scene	62,87(7)	68,17(5)	82,38(1)	66,43(6)	68,73(4)	69,75(2)	69,52 (3)
Slashdot	44,21(4)	45,96(3)	70,37(1)	39,93(5)	5,69(7)	7,67(6)	53,18 (2)
Yeast	58,86(6)	59,38(5)	60,93(2)	59,40(4)	56,89(7)	60,41(3)	61,84 (1)
Prom. rank.	(5,14)	(3,32)	(1,86)	(4,59)	(6,36)	(4,18)	(2,55)
$\boldsymbol{F_1}$	BR	STA	STA^{y}	ECC*	MLkNN	IBLR	RAkEL
Bibtex	37,02(4)	37,92(2)	37,54(3)	36,86(5)	16,98(7)	22,38(6)	41,28 (1)
Emotions	49,20(6)	50,33(4)	60,87(2)	49,81(5)	41,60(7)	62,97(1)	51,95 (3)
Enron	55,66(3)	57,78(2)	58,20(1)	51,14(5)	41,82(6)	41,52(7)	52,43 (4)
Genbase	99,18(3,5)	99,10(5)	99,21(2)	99,18(3,5)	95,81(7)	98,78(6)	99,50 (1)
Image	42,12(6)	42,38(5)	51,68(1)	43,46(4)	40,63(7)	44,91(3)	46,32 (2)
Mediamill	59,17(3)	47,06(7)	57,23(6)	58,15(4)	59,55(2)	60,17(1)	57,72 (5)
Medical	77,33(5)	79,45(3)	79,82(1)	78,83(4)	59,41(7)	62,19(6)	79,65 (2)
Reuters	84,13(5)	84,21(4)	87,05(2)	84,69(3)	80,50(6)	69,10(7)	88,58 (1)
Scene	61,25(7)	66,75(5)	68,46(4)	66,31(6)	68,49(3)	69,97(1)	68,84 (2)
Slashdot	44,33(4)	46,05(3)	55,47(1)	40,66(5)	5,84(7)	7,73(6)	50,49 (2)
Yeast	61,68(5)	61,86(3)	60,98(6)	61,80(4)	60,97(7)	62,85(1)	62,48 (2)
Prom. rank.	(4,68)	(3,91)	(2,64)	(4,41)	(6,00)	(4,09)	(2,27)
Acierto	BR	STA	STA^{y}	ECC*	MLkNN	IBLR	RAkEL
Bibtex	31,50(4)	32,15(3)	32,32(2)	31,46(5)	13,61(7)	18,09(6)	34,28 (1)
Emotions	42,27(6)	43,46(3)	51,76(2)	43,24(4)	34,09(7)	55,08(1)	42,98 (5)
Enron	44,69(3)	45,95(2)	47,09(1)	39,68(5)	31,83(7)	31,99(6)	41,30 (4)
Genbase	98,94(3,5)	98,82(5)	98,97(2)	98,94(3,5)	94,86(7)	98,25(6)	99,29 (1)
Image	38,60(6)	38,87(5)	47,32(1)	40,15(4)	38,45(7)	42,46(2)	42,15 (3)
Mediamill	46,70(3)	34,09(7)	45,42(4)	44,69(6)	48,11(2)	48,82(1)	45,10 (5)
Medical	74,51(5)	75,43(4)	76,95(1)	76,33(3)	56,76(7)	58,19(6)	76,88 (2)
Reuters	81,67(5)	81,76(4)	84,00(2)	82,41(3)	78,11(6)	67,10(7)	86,38 (1)
Scene	59,41(7)	64,88(5)	64,00(6)	65,05(4)	67,03(3)	68,77(1)	67,28 (2)
Slashdot	42,71(4)	44,29(3)	49,70(1)	39,32(5)	5,68(7)	7,42(6)	47,18 (2)
Yeast	50,71(5)	50,93(4)	49,68(7)	50,96(3)	50,50(6)	52,65(1)	51,75 (2)
Prom. rank.	(4,68)	(4,09)	(2,64)	(4,14)	(6,00)	(3,91)	(2,55)

Tabla 3. Resultados de todos los sistemas para pérdida *Hamming* y error 0/1

Hamming	BR	STA	STA^{y}	ECC*	MLkNN	IBLR	RAkEL
Bibtex	1,21(2)	1,26(4)	1,21(2)	1,21(2)	1,36(5)	1,60(7)	1,49 (6)
Emotions	22,03(4)	21,83(3)	23,15(5)	21,72(2)	26,21(6)	18,72(1)	28,01 (7)
Enron	4,46(1)	4,83(3)	4,88(4)	4,72(2)	5,22(5)	5,60(6)	5,85 (7)
Genbase	0,08(3,5)	0,09(5)	0,07(2)	0,08(3,5)	0,45(7)	0,19(6)	0,06 (1)
Image	20,25(5)	20,23(4)	21,61(6)	19,80(3)	19,28(2)	18,75(1)	24,40 (7)
Mediamill	2,76(2)	5,50(7)	3,10(6)	2,86(5)	2,70(1)	2,82(4)	2,81 (3)
Medical	0,99(4)	1,12(5)	0,98(3)	0,95(1,5)	1,56(6)	1,90(7)	0,95 (1,5)
Reuters	4,58(3)	4,60(4)	5,77(5)	4,42(2)	6,03(6)	8,32(7)	3,85 (1)
Scene	9,83(4)	9,85(5)	18,31(7)	9,06(3)	8,66(2)	8,38(1)	9,90 (6)
Slashdot	3,73(1)	3,86(3)	8,80(7)	3,78(2)	5,18(6)	5,17(5)	4,65 (4)
Yeast	19,81(3)	19,85(4)	21,56(7)	19,98(5)	19,43(2)	19,18(1)	20,30 (6)
Prom. rank.	(2,95)	(4,27)	(4,91)	(2,82)	(4,36)	(4,18)	(4,50)
0/1	BR	STA	STA^{y}	ECC*	MLkNN	IBLR	RAkEL
Bibtex	82,83(3)	82,99(4)	81,54(1)	82,61(2)	94,06(7)	91,64(6)	83,56 (5)
Emotions	79,42(5)	77,91(4)	74,70(2)	77,05(3)	87,00(7)	68,97(1)	83,45 (6)
Enron	86,90(2)	88,07(3)	85,25(1)	93,07(6)	94,89(7)	92,30(5)	88,49 (4)
Genbase	1,81(3)	2,11(5)	1,81(3)	1,81(3)	8,16(7)	4,08(6)	1,51 (1)
Image	71,50(7)	71,25(6)	65,20(2)	69,40(4)	67,95(3)	64,70(1)	69,70 (5)
Mediamill	90,36(4)	97,90(7)	88,14(3)	93,80(6)	86,24(2)	85,86(1)	91,14 (5)
Medical	33,94(4)	36,61(5)	31,49(3)	31,19(1)	51,12(6)	52,98(7)	31,39 (2)
Reuters	25,69(5)	25,54(4)	24,60(3)	24,43(2)	29,04(6)	38,88(7)	20,24 (1)
Scene	46,03(7)	40,63(5)	44,99(6)	38,72(4)	37,35(2,5)	34,82(1)	37,35 (2,5)
Slashdot	61,95(2)	60,82(1)	62,85(4)	64,57(5)	94,76(7)	93,47(6)	62,11 (3)
Yeast	84,53(7)	83,95(5)	84,07(6)	83,58(4)	82,29(2)	79,19(1)	83,08 (3)
Prom. rank.	(4,45)	(4,45)	(3,09)	(3,64)	(5,14)	(3,82)	(3,41)

paraciones por pares mediante el test de Bergmann-Hommel, usando el código proporcionado en [7].

Observando ambas tablas, nuestra propuesta supera al método original en todas las medidas excepto en *Hamming*. Si consideramos todos los algoritmos, encontramos que nuestro método logra los mejores resultados en tres de las medidas (Precisión, Exhaustividad y error 0/1), es segundo en dos (F_1 y Acierto), detrás en ambos casos del algoritmo RAkEL, y el peor en pérdida *Hamming*, donde el mejor método es ECC*. El método STA no alcanza las dos primeras posiciones en ninguna medida y el BR solo en *Hamming*. Todo esto indica que nuestro método detecta mejor la etiquetas relevantes, como muestra la Exhaustividad, pero tiende a añadir algunas etiquetas no relevantes, como indican los errores en *Hamming*; pero el compromiso es bueno mirando el valor de F_1.

De todas formas, en los experimentos se observan muy pocas diferencias significativas entre todos los algoritmos. En concreto en Precisión, pérdida *Hamming* y error 0/1 no hay ninguna diferencia significativa entre ningún par de métodos. En F_1 y Acierto solamente se observa que RAkEL y nuestro método son mejores al 95 % que MLkNN. La única medida en la que se aprecian más diferencias es en Exhaustividad, donde nuestro método es significativamente mejor que ECC*, MLkNN y BR al 95 %. Además, RAkEL es significativamente mejor que MLkNN y BR al 95 % y el STA es mejor que MLkNN también al 95 %.

5. Conclusiones

En este artículo se ha presentado una variante de los modelos apilados para la clasificación multi-etiqueta que permite mejorar su rendimiento en varias de las medidas más utilizadas en este tipo de tarea de aprendizaje. La idea fundamental del método propuesto consiste en tratar de detectar la dependencia condicional entre todas las etiquetas. Para ello se construyen modelos que tienen en cuenta no solamente la descripción del objeto, sino también la información del resto de etiquetas contenida en el conjunto de entrenamiento. Los resultados experimentales efectuados sobre once conjuntos de datos demuestran que nuestro método obtiene buenos resultados en aquellas medidas que premian predecir correctamente las etiquetas relevantes. La única desventaja observada es que tiende a predecir como relevante alguna etiqueta que no lo es.

Referencias

1. E. Alvares, J. Metz, and M. Monard. A Simple Approach to Incorporate Label Dependency in Multi-label Classification. In *Advances in Soft Computing*, volume 6438 of *Lecture Notes in Computer Science*, chapter 3, pages 33–43. Springer, 2010.
2. W. Cheng and E. Hüllermeier. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2-3):211–225, 2009.
3. A. Clare and R. D. King. Knowledge discovery in multi-label phenotype data. In *European Conf. on Data Mining and Knowledge Discovery*, pages 42–53, 2001.
4. K. Dembczynski, W. Cheng, and E. Hüllermeier. Bayes Optimal Multilabel Classification via Probabilistic Classifier Chains. In *ICML*, pages 279–286, 2010.
5. A. Elisseeff and J. Weston. A Kernel Method for Multi-Labelled Classification. In *ACM Conf. on Research and Develop. in Infor. Retrieval*, pages 274–281, 2005.
6. J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, and K. Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73:133–153, 2008.
7. S. García and F. Herrera. An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.
8. S. Godbole and S. Sarawagi. Discriminative methods for multi-labeled classification. In *Pacific-Asia Conf. on Know. Disc. and Data Mining*, pages 22–30, 2004.
9. C.-J. Lin, R. C. Weng, and S. S. Keerthi. Trust region Newton method for logistic regression. *Journal of Machine Learning Research*, 9(Apr):627–650, 2008.
10. J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. In *ECML'09*, LNCS, pages 254–269. Springer, 2009.
11. G. Tsoumakas, A. Dimou, E. Spyromitros, V. Mezaris, I. Kompatsiaris, and I. Vlahavas. Correlation-based pruning of stacked binary relevance models for multi-label learning. In *Workshop on learning from multi-label data*, pages 101–116, 2009.
12. G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, pages 667–685. 2010.
13. G. Tsoumakas and I. Vlahavas. Random k-Labelsets: An Ensemble Method for Multilabel Classification. In *European Conference on Machine Learning and Knowledge Discovery in Databases*, LNCS, pages 406–417. Springer, 2007.
14. D. H. Wolpert. Stacked generalization. *Neural Networks*, 5:214–259, 1992.
15. M.-L. Zhang and Z.-H. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.