

Combinación de vistas para clasificación multi-etiqueta: estudio preliminar

Eva L. Gibaja, Jose M. Moyano, y Sebastián Ventura

Universidad de Córdoba. Departamento de Informática y Análisis Numérico

Resumen La clasificación multi-etiqueta con múltiples vistas de los datos es un campo de investigación poco explorado. En este trabajo proponemos un enfoque basado en *ensembles* que permite la fusión de vistas durante el proceso de decisión. El estudio preliminar realizado muestra que nuestra propuesta, además de mejorar los resultados producidos por las vistas individuales, mejora, en tiempo y resultados, al enfoque clásico de concatenar las vistas en un único conjunto de atributos.

Keywords: Clasificación multi-etiqueta, aprendizaje multi-vista, *ensembles*

1. Introducción

La clasificación multi-etiqueta (*Multi-Label Classification*, MLC) es un paradigma de aprendizaje supervisado que ha experimentado un gran crecimiento estos últimos años debido a su éxito en problemas de actualidad como la clasificación de textos y multimedia, la predicción de funciones de genes y proteínas, el *marketing* directo, o la minería de redes sociales [4].

Estas aplicaciones tienen en común que un objeto (patrón o instancia) puede pertenecer a más de una clase (etiqueta) simultáneamente, no satisfaciéndose la restricción *only-one-label-per-pattern* del aprendizaje clásico (*single-label*). Para representar este hecho, las etiquetas son variables binarias que indican la pertenencia a cada una de las clases de interés de forma análoga al aprendizaje multiclase, pero con la diferencia de que un ejemplo puede tener más de un valor binario activo. La MLC implica mayor dificultad debido a aspectos como el coste computacional de generar/consultar los modelos, la presencia de etiquetas desbalanceadas y relaciones entre etiquetas o la gran dimensionalidad de los datos (número de patrones, atributos y etiquetas). La gran dimensionalidad del espacio de salida es considerada el mayor reto de la MLC [20].

En numerosos dominios de aplicación un objeto tiene diferentes representaciones y es descrito por varios conjuntos de atributos (o vistas) obtenidos a partir de diversas fuentes o de varios extractores de características. El resultado es que las variables descriptoras (el espacio de entrada) se pueden particionar. Así, por ejemplo, un fragmento musical puede ser descrito tanto con atributos de ritmo como de timbre.

A pesar de que los datos del mundo real se encuentran almacenados de forma distribuida, particionada y con una organización heterogénea en múltiples fuentes de información, los algoritmos de aprendizaje tradicionales infieren modelos

a partir de un único conjunto de datos homogéneo que concatena o fusiona los distintos conjuntos que lo integran. Esta metodología no funciona bien en todos los casos, pudiendo aumentar la ambigüedad del conjunto de datos y dificultar el proceso de aprendizaje. También se puede añadir que algunos trabajos citan mejoras de rendimiento en problemas en los que no existe una partición natural de los atributos en vistas mediante la utilización de particiones artificiales [13]. El aprendizaje multi-vista (*Multi-View Learning*, MVL) pretende solucionar este tipo de problemas, combinando información heterogénea y complementaria para aprender modelos que obtengan mejores resultados que modelos independientes generados a partir de informaciones independientes [16].

En este trabajo presentamos una propuesta para clasificación multi-etiqueta con múltiples vistas (*Multi-View Multi-Label*, MVML). Este enfoque de aprendizaje más flexible permite que cada patrón esté representado por varios conjuntos de atributos y, además, que esté asociado simultáneamente a varias etiquetas. La propuesta está basada en *ensembles* y fusiona las vistas durante el proceso de decisión. Nuestro objetivo es generar un modelo para cada vista y combinar los resultados obtenidos mediante un esquema de votación. Realizaremos un estudio preliminar para determinar si los resultados obtenidos por esta combinación de hipótesis individuales supera, no solo la precisión de cada vista, sino además la precisión del modelo generado mediante la combinación de todas las vistas.

El resto del artículo se organiza de la siguiente manera: la Sección 2 describe las principales aportaciones realizadas hasta la fecha tanto para MLC, como para MVL y MVML. La Sección 3 realiza una descripción formal del problema y presenta nuestra propuesta. A continuación, la Sección 4 describe el marco experimental. La Sección 5 analiza los resultados obtenidos. La Sección 6 finaliza con las principales conclusiones y líneas de trabajo futuro.

2. Antecedentes

Los problemas de MLC pueden abordarse desde dos puntos de vista [4]: *transformación de problemas y adaptación de algoritmos*. El primero se basa en transformar un problema multi-etiqueta en uno o varios problemas de una sola etiqueta que posteriormente son resueltos utilizando un algoritmo clásico de clasificación. El segundo enfoque consiste en adaptar o extender algún algoritmo clásico de clasificación de modo que trabaje directamente con datos multi-etiqueta.

Podemos distinguir un conjunto de métodos de transformación que consideran las combinaciones de etiquetas presentes en el conjunto de datos como una nueva etiqueta de tipo multiclase. Este es el caso de *Label Powerset* (LP) [4]. Otro grupo de métodos realiza una descomposición *pairwise* generando un modelo binario para cada par de etiquetas [2]. Cabe destacar un tercer grupo de métodos de transformación que descomponen el problema multi-etiqueta en un problema binario por cada etiqueta. Este es el caso de *Binary Relevance* (BR) [4], un algoritmo de referencia debido a que, aunque asume la independencia entre las etiquetas, escala de forma lineal con el número de etiquetas manteniendo buenos resultados. Por otro lado, la mayoría de los algoritmos clásicos han sido

abordados desde el enfoque de adaptación. En [4] se pueden encontrar, entre otras, propuestas basadas en árboles decisión, SVMs y basadas en instancias.

Los algoritmos de MVL se pueden categorizar en tres grupos: algoritmos de co-aprendizaje, que entrenan alternativamente vistas de datos para maximizar la coincidencia en la salida [1], algoritmos basados en combinaciones de funciones *kernel* [5] y algoritmos de aprendizaje en subespacios latentes de características compartidas por las diferentes vistas [11]. El MVL se ha estudiado en numerosas aplicaciones, pero la mayoría de los métodos existentes aprenden una única tarea.

Las propuestas MVML, que combinan MLV y MLC, son recientes. En [6] se propone un método basado en grafos para hacer el mejor uso de la información entre múltiples vistas. Además se desarrolla un proceso iterativo para optimizar el modelo propuesto. En [19] se utilizó un método de co-regularización para obtener funciones consistentes en los ejemplos sin etiquetar y que a la vez consiguen bajos errores de predicción en los ejemplos etiquetados. Finalmente, en [7] se aprenden las estructuras predictivas compartidas por varias vistas y múltiples tareas y se utiliza la consistencia entre las distintas vistas para mejorar el rendimiento. Respecto a métodos basados en multi-clasificadores cabe destacar las versiones MVML para *boosting* y *bagging* [21] que aprenden, para cada vista, un *ensemble* formado por varios clasificadores base (en este caso SVM).

3. MVMLE: *ensemble* para clasificación MVML

En esta sección describimos formalmente la clasificación MVML y desarrollamos una propuesta de fusión de vistas para MLMV basada en *ensembles*.

3.1. Representación de los datos

Para la tarea de clasificación MVML consideraremos un espacio de salida de q etiquetas, $\mathcal{Y} = \{\lambda_1, \lambda_2, \dots, \lambda_q\} | q > 1$. Cada subconjunto $Y \subseteq \mathcal{Y}$ se denomina *labelset*. Cada vista, j , está descrita por un espacio de entrada de d_j atributos, \mathcal{X}^j . Así, cada $\mathbf{x}^j = (x_1, \dots, x_{d_j}) \in \mathcal{X}^j$ es un vector de atributos para la vista j . Consideraremos que $D = \{(\mathbf{x}_i^1, \dots, \mathbf{x}_i^v, Y_i) | 1 \leq i \leq m\}$ es un conjunto de datos de m objetos representados mediante v vistas. Las asociaciones de etiquetas también se pueden representar mediante un vector binario q -dimensional, $\mathbf{y} = (y_1, y_2, \dots, y_q) = \{0, 1\}^q$, en el que un elemento vale 1 si la etiqueta es relevante y 0 en otro caso.

3.2. Combinación de vistas para MVML

En este trabajo consideraremos dos esquemas básicos para la combinación de vistas en aprendizaje *single-label* descritos en [12]. Por un lado una fusión temprana (*early fusion*) de las vistas en la que todos los atributos se combinan en un único vector de características y posteriormente se aplica el proceso de aprendizaje. En la práctica equivale a no trabajar con vistas (ver Figura 1(a)). Uno de los problemas de este enfoque es tratar con múltiples vistas heterogéneas

Algoritmo 1 MVMLE entrenamiento

Input: $D = \{(\mathbf{x}_i^1, \dots, \mathbf{x}_i^v, Y_i) | 1 \leq i \leq m\}$ m ejemplos representados por v vistas y q etiquetas

- 1: **for** $j \leftarrow 1$ to v **do**
- 2: {Generar un clasificador multi-etiqueta para cada vista, j }
- 3: $h_j : \mathcal{X}^j \rightarrow 2^{\mathcal{Y}}$
- 4: **end for**

Output: $H : \mathcal{X}^1 \times \dots \times \mathcal{X}^v \rightarrow 2^{\mathcal{Y}} = \{h_j | 1 \leq j \leq v\}$

que, a menudo, pueden resultar difíciles de combinar. La ventaja es, que al realizarse la fusión a nivel de características, permite explotar las relaciones entre los atributos de las vistas. Por otro lado, consideraremos una fusión tardía (*late fusion*) que se aplica en la etapa de decisión. Inicialmente se toman decisiones locales basadas en características individuales y, posteriormente, dichas decisiones son combinadas para obtener la decisión final (ver Figura 1(b)). Este método se puede paralelizar, pero no tiene en cuenta las relaciones entre los atributos de las vistas.

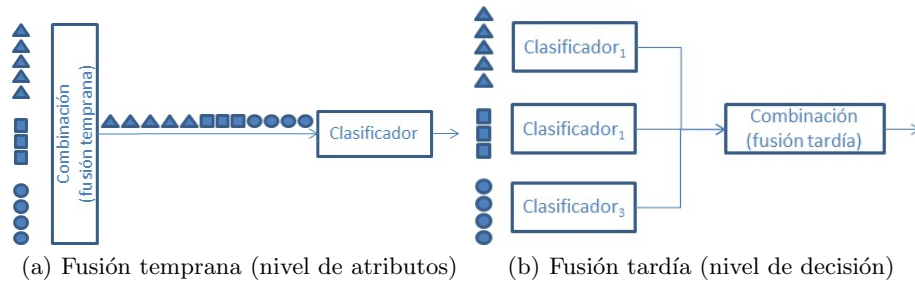


Figura 1. Enfoques para fusión de vistas

Nuestra propuesta, a la que hemos denominado *Multi-View Multi-Label Ensemble* (MVMLE), se basa en aplicar el enfoque de fusión tardía. La idea fundamental es desarrollar en la etapa de entrenamiento un modelo de *ensemble* formado por un clasificador base multi-etiqueta para cada vista. El proceso de entrenamiento está descrito en el Algoritmo 1 y permite utilizar cualquier algoritmo base, siempre que éste sea multi-etiqueta. Una vez generado el *ensemble*, el modelo está listo para predecir el conjunto de clases que corresponden a un ejemplo sin etiquetar conforme al proceso descrito en el Algoritmo 2. Las decisiones de los clasificadores base se combinan mediante un esquema de votación en el que se contabilizan los votos obtenidos para cada etiqueta y se promedian dividiendo entre el número de clasificadores base (el número de vistas). Si el valor obtenido supera un cierto *umbral* (que es el único parámetro del algoritmo) la salida para dicha etiqueta será positiva.

Algoritmo 2 MVMLE predicción

Input: $umbral$, $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^v)$ un objeto sin etiquetar representado por v vistas

```

1: {Inicializar el total de votos para cada etiqueta}
2: for  $i \leftarrow 1$  to  $q$  do
3:    $votos_i \leftarrow 0$ 
4: end for
5: for  $j \leftarrow 1$  to  $v$  do
6:   {Realizar la predicción para cada vista}
7:    $\mathbf{y}_j \leftarrow h_j(\mathbf{x}^j)$ 
8:   {Contabilizar los votos para cada etiqueta}
9:   for  $i \leftarrow 1$  to  $q$  do
10:    if  $\mathbf{y}_j[i] \neq 0$  then
11:       $votos_i \leftarrow votos_i + 1$ 
12:    end if
13:  end for
14: end for
15: {La salida será positiva si se supera el umbral}
16: for  $i \leftarrow 1$  to  $q$  do
17:   if  $votos_i/v \geq umbral$  then
18:      $\mathbf{y}[i] \leftarrow 1$ 
19:   else
20:      $\mathbf{y}[i] \leftarrow 0$ 
21:   end if
22: end for

```

Output: $\mathbf{y} = (y_1, y_2, \dots, y_q) = \{0, 1\}^q$

4. Experimentos

Esta sección describe el proceso experimental realizado, las métricas de evaluación y conjuntos de datos utilizados.

4.1. Conjuntos de datos

Para evaluar el rendimiento MVMLE hemos utilizado 4 conjuntos de datos procedentes de diversos dominios de aplicación: *emotions* [14], *yeast* [3], *plant* [17] y *human* [17]. Han sido obtenidos de los repositorios MULAN [15] y LABIC [18] y son variados en complejidad, número de etiquetas, número de atributos y número de ejemplos. La Tabla 1 resume sus principales características. Han sido ordenados conforme al criterio de complejidad definido en [9] que consiste en el producto de $labels \times instances \times features$ (lo denominaremos LIF). La cardinalidad es el número medio de etiquetas en los ejemplos y la densidad es la cardinalidad dividida por el total de etiquetas, q . El *distinct* es el número de combinaciones de etiquetas diferentes que hay en el conjunto de datos.

Emotions tiene 72 características musicales para 593 canciones categorizadas en hasta 6 tipos de emoción. Hemos considerado 2 vistas. La primera tiene 8 atributos sobre ritmo obtenidas extrayendo cambios periódicos de un histograma

Dataset	Dominio	Vistas	Atributos	Pat.	Etiq.	Card.	Dens.	Dist.	LIF
Emotions	Música	2	8+64	593	6	1.868	0.311	27	2.56E+5
Yeast	Biología	2	79+24	2417	14	4.237	0.302	198	3.48E+6
Plant	Biología	3	20+20+400	978	12	1.078	0.089	32	5.16E+6
Human	Biología	3	20+20+400	3106	14	1.185	0.084	85	1.91E+7

Tabla 1. Descripción de los conjuntos de datos

temporal o histograma de ritmo (*beat histogram*). La segunda vista tiene 64 atributos sobre timbre basados en *Mel Frequency Cepstral Coefficients*, MFCCs y en el centroide, *rolloff* y flujo espectral. *Yeast* tiene 2417 patrones, cada uno de los cuales es la concatenación de la expresión genética (79 atributos) y el perfil filogenético de un gen (24 atributos). *Plant* y *human* son dos conjuntos de datos para detectar la localización subcelular (*subcellular locations*) de proteínas en base a sus secuencias. Tienen 978 y 3106 secuencias de proteínas de plantas y humanos respectivamente. Sus 440 atributos se han particionado en 3 vistas correspondientes a la composición de amino ácidos (20 atributos), pseudo amino ácidos (20 atributos) y dipéptidos (400 atributos). *Plant* tiene 12 etiquetas y *human* 14, correspondientes a las localizaciones de las proteínas en la célula.

Los datos han sido particionados en 5 *folds* utilizando el procedimiento para estratificado específico multi-etiqueta descrito en [10]. Cada combinación de etiquetas distinta en el conjunto de datos se considera que es una clase y la estratificación se realiza sobre estas nuevas clases. Se pretende conseguir que la distribución de las distintas combinaciones de etiquetas sea la misma en todas las particiones.

4.2. Descripción de los experimentos

En este estudio preliminar analizaremos si los resultados obtenidos mediante la fusión tardía de MVMLE supera, por un lado, a los de las vistas individuales y, por otro, a los de la fusión temprana. Como se ha indicado en la Sección 3, MVMLE es independiente del algoritmo base utilizado, siempre y cuando sea de tipo multi-etiqueta. De forma particular, y debido a su simplicidad y eficiencia, hemos utilizado en todos los casos BR como algoritmo base. El uso de J48 como clasificador base para los clasificadores binarios de BR es prácticamente un estándar en MLC. El único parámetro de MVMLE es el valor del *umbral* para la votación, que hemos establecido en 0,5. La implementación de MVMLE se ha realizado en Java utilizando la librería MULAN [15] que proporciona un conjunto de clases base para desarrollo de algoritmos multi-etiqueta. La implementación de BR también está disponible en esta librería. Finalmente, los conjuntos de datos se encuentran en formato *arff* con las vistas combinadas en una única vista (conforme al enfoque de fusión temprana). Para procesar cada vista de forma independiente hemos desarrollado un *wrapper* en Java.

4.3. Métricas de evaluación

Para cada conjunto de datos y algoritmo se ha registrado el tiempo de ejecución (*train* más *test*) en milisegundos y se han obtenido 27 métricas de evaluación para aprendizaje multi-etiqueta, *basadas en ejemplos* y *basadas en etiquetas*, definidas en [4]. Las basadas en ejemplos se calculan para cada instancia y después se promedian. Las basadas en etiquetas se calculan a partir de los valores de las tablas de contingencia de cada una de las etiquetas binarias del problema. Al haber una tabla de contingencia por etiqueta, se pueden agregar siguiendo un enfoque *macro* (que calcula la métrica binaria para cada etiqueta y finalmente promedia el valor) o siguiendo un enfoque *micro* (que primero agrega/suma los valores de todas las tablas y luego calcula el valor de la métrica). Las métricas *mean average precision* (MAP), *geometric mean average precision* (GMAP), *mean average interpolated precision* (MAIP) y *geometric mean average interpolated precision* (GMAIP), están basadas en etiquetas con promedio macro [8].

5. Resultados y discusión

La Tabla 2 presenta los resultados del estudio preliminar realizado. Para cada conjunto de datos se recogen el tiempo de ejecución y los valores de las 27 métricas para: a) las vistas individuales (columna *Vista_i*), b) el enfoque de fusión temprana (columna *Concat*) y c) el enfoque de fusión tardía (columna *MVMLE*). Las métricas están precedidas por un símbolo \downarrow ó \uparrow que indica si se minimizan o maximizan. Se ha marcado en negrita el mejor resultado considerando las columnas *Concat* y *MVMLE*. Se han subrayado aquellos casos en que las vistas individuales mejoran tanto el enfoque de fusión temprana como tardía.

Respecto a *emotions*, MVMLE supera a la fusión temprana en 18 de las 27 métricas con un tiempo de ejecución muy similar. En todos estos casos MVMLE supera además los resultados obtenidos por las vistas individuales. La fusión temprana mejora a MVMLE en 9 de 27 métricas, pero en 3 de estos casos (*specificity* macro, micro y basada en ejemplos) no consigue mejorar los resultados de la *vista₂* y en 1 caso (*subset accuracy*) no mejora los resultados de la *vista₁*.

En el conjunto de datos *yeast* el tiempo de ejecución de MVMLE mejora al de la fusión temprana. Consideramos que es debido a que, al trabajar con varias vistas, se reducen la complejidad del espacio de entrada y el tiempo de ejecución. En 24 de 27 métricas los resultados obtenidos por MVMLE mejoran a la fusión temprana. De estos 24 casos, solamente en 5 alguna de las vistas ofrece mejores resultados. La fusión temprana solo supera a MVMLE en 3 de 27 métricas, y en estos casos no consigue mejorar los resultados de ninguna vista individual.

Respecto a *plant*, el tiempo de ejecución de MVMLE es menor que el de la fusión temprana. MVMLE obtiene mejores resultados en 17 de las 27 métricas y siempre supera los resultados obtenidos por las vistas individuales. La fusión temprana supera a MVMLE en 10 casos, pero en 7 de estos no consigue mejorar los resultados de alguna vista. Se observa además que, en los casos en que MVMLE obtiene peores resultados que la fusión temprana, la *vista₂* tiene malos

	Emotions				Yeast				Plant				Human					
	Vista ₁	Vista ₂	Concat	MVMLE	Vista ₁	Vista ₂	Concat	MVMLE	Vista ₁	Vista ₂	Vista ₃	Concat	MVMLE	Vista ₁	Vista ₂	Vista ₃	Concat	MVMLE
↓Time _{ms}	96	10	100	105	971	225	1367	996	74	41	2618	2839	2709	287	220	14990	16024	15166
↓Hamming loss	0.252	0.298	0.249	0.286	0.243	0.223	0.251	0.241	0.102	0.096	0.135	0.134	0.092	0.091	0.088	0.123	0.120	0.085
↓Subset accuracy	0.177	0.116	0.175	0.150	0.065	0.076	0.060	0.072	0.141	0.042	0.110	0.105	0.085	0.137	0.041	0.100	0.118	0.087
↑Precision _{EB}	0.559	0.448	0.576	0.545	0.625	0.671	0.596	0.606	0.168	0.042	0.196	0.181	0.091	0.186	0.052	0.217	0.233	0.109
↑Recall _{EB}	0.581	0.374	0.593	0.706	0.518	0.533	0.579	0.660	0.180	0.043	0.265	0.245	0.089	0.177	0.047	0.291	0.303	0.100
↑Fmeasure _{EB}	0.536	0.384	0.548	0.581	0.536	0.566	0.557	0.604	0.170	0.043	0.213	0.199	0.089	0.177	0.048	0.232	0.248	0.103
↑Accuracy _{EB}	0.447	0.313	0.453	0.472	0.417	0.447	0.434	0.481	0.163	0.042	0.185	0.174	0.088	0.166	0.046	0.196	0.212	0.099
↑Specificity _{EB}	0.826	0.851	0.831	0.721	0.866	0.890	0.829	0.811	0.970	0.989	0.924	0.928	0.990	0.978	0.992	0.992	0.934	0.991
↑Precision _{mic}	0.596	0.529	0.590	0.712	0.617	0.664	0.577	0.656	0.172	0.040	0.264	0.241	0.085	0.167	0.044	0.283	0.293	0.094
↑Fmeasure _{mic}	0.592	0.435	0.596	0.608	0.519	0.534	0.577	0.622	0.232	0.067	0.260	0.243	0.142	0.237	0.077	0.280	0.292	0.157
↑Recall _{mic}	0.819	0.849	0.823	0.714	0.860	0.883	0.581	0.804	0.970	0.989	0.924	0.928	0.990	0.978	0.992	0.934	0.991	0.036
↑Specificity _{mic}	0.719	0.667	0.708	0.779	0.704	0.773	0.685	0.797	0.693	0.738	0.560	0.517	0.773	0.778	0.795	0.566	0.560	0.819
↑Auc _{mic}	0.596	0.497	0.597	0.529	0.394	0.454	0.401	0.403	0.174	0.049	0.169	0.188	0.187	0.169	0.060	0.138	0.156	0.179
↑Precision _{mac}	0.576	0.344	0.585	0.689	0.317	0.309	0.388	0.427	0.103	0.018	0.164	0.170	0.044	0.079	0.016	0.143	0.148	0.036
↑Fmeasure _{mac}	0.576	0.360	0.586	0.585	0.334	0.327	0.389	0.399	0.120	0.025	0.163	0.175	0.066	0.099	0.024	0.140	0.149	0.055
↑Specificity _{mac}	0.813	0.834	0.819	0.700	0.776	0.787	0.705	0.746	0.965	0.987	0.916	0.921	0.988	0.973	0.990	0.922	0.925	0.988
↑Auc _{mac}	0.711	0.623	0.704	0.751	0.565	0.573	0.579	0.603	0.559	0.504	0.524	0.523	0.567	0.544	0.513	0.526	0.522	0.565
↑Fmeasure _{mac}	0.576	0.360	0.586	0.585	0.334	0.327	0.389	0.399	0.120	0.025	0.163	0.175	0.066	0.099	0.024	0.140	0.149	0.055
↑Specificity _{mac}	0.813	0.834	0.819	0.700	0.776	0.787	0.705	0.746	0.965	0.987	0.916	0.921	0.988	0.973	0.990	0.922	0.925	0.988
↑Avg. precision	0.703	0.647	0.697	0.739	0.629	0.692	0.613	0.716	0.485	0.489	0.388	0.367	0.519	0.538	0.511	0.401	0.414	0.541
↓Coverage	2.435	2.601	2.528	2.135	9.091	7.720	9.309	7.204	3.539	2.966	5.049	5.642	2.629	3.277	2.999	6.079	6.106	2.661
↓One error	0.408	0.496	0.400	0.359	0.413	0.351	0.425	0.275	0.695	0.713	0.772	0.777	0.679	0.643	0.683	0.731	0.710	0.649
↓Is error	0.658	0.757	0.681	0.616	0.903	0.854	0.916	0.843	0.724	0.722	0.796	0.799	0.705	0.677	0.709	0.787	0.763	0.690
↓Error set size	2.008	2.395	2.076	1.607	11.238	8.520	12.036	7.575	3.582	3.025	5.190	5.810	2.664	3.316	3.038	6.415	6.307	2.693
↓Ranking loss	0.278	0.330	0.287	0.226	0.298	0.229	0.317	0.205	0.305	0.258	0.442	0.497	0.226	0.223	0.205	0.425	0.427	0.179
↑MAP	0.543	0.447	0.539	0.588	0.364	0.383	0.376	0.411	0.170	0.113	0.149	0.156	0.190	0.143	0.105	0.118	0.125	0.148
↑GMAP	0.530	0.438	0.527	0.574	0.260	0.275	0.269	0.283	0.117	0.081	0.106	0.111	0.129	0.075	0.057	0.064	0.068	0.075
↑MAIP	0.585	0.488	0.581	0.623	0.395	0.416	0.408	0.438	0.191	0.130	0.170	0.176	0.211	0.162	0.122	0.137	0.147	0.167
↑GMaip	0.575	0.480	0.571	0.610	0.290	0.307	0.300	0.325	0.132	0.094	0.120	0.125	0.145	0.087	0.067	0.075	0.082	0.087

Tabla 2. Resultados experimentales

resultados para la métrica en cuestión. Al combinar los resultados de las vistas por votación, MVMLE es más sensible a rendimientos bajos en alguna vista que la fusión temprana, que es capaz de considerar todos los atributos a la vez.

En el caso de *human* el tiempo de ejecución de MVMLE mejora al de la fusión temprana y obtiene mejores resultados en 18 de las 27 métricas. En 7 de estos casos alguna vista individual produce mejores resultados que el *ensemble*. La fusión temprana mejora tanto a las vistas individuales como a MVMLE en 9 ocasiones. También, igual que ocurría con *plant*, se observa que cuando la fusión temprana mejora a MVMLE, la *vista₂* tiene malos resultados para la métrica, lo que influye negativamente en la votación.

6. Conclusiones y trabajo futuro

Hemos realizado un estudio preliminar de dos enfoques para la fusión de vistas en clasificación multi-etiqueta y hemos presentado MVMLE, una propuesta para la fusión de vistas a nivel de decisión. El estudio previo realizado con 4 conjuntos de datos muestra que la fusión tardía de MVMLE reduce la complejidad del problema, y por tanto el tiempo de ejecución respecto a la fusión temprana. MVMLE mejora en la mayoría de las 27 métricas estudiadas los resultados de la fusión temprana a la vez que mejora los resultados obtenidos para las vistas individuales. Al combinar las vistas en la etapa de decisión, MVMLE permite que los clasificadores base puedan ser generados en paralelo, mientras que con la fusión temprana esto es posible. El inconveniente de MVMLE es que, al combinar los resultados de las vistas a nivel de decisión, no se consideran las relaciones entre todos los atributos del problema, lo que lo hace más sensible cuando alguna de las vistas produce malos resultados.

Los resultados del estudio son positivos y nos llevan a plantearnos analizar si con otros algoritmos multi-etiqueta MVMLE se comporta de forma similar. También pretendemos establecer mecanismos para ponderar la votación de los clasificadores base de modo que malos resultados para una vista no influyan de forma tan negativa en la decisión final.

Agradecimientos

Esta investigación ha sido financiada por el proyecto TIN2014-55252-P del Ministerio de Economía y Competitividad y fondos FEDER.

Referencias

1. Amini, M.R., Goutte, C.: A co-classification approach to learning from multilingual corpora. *Machine learning* 79(1-2), 105–121 (2010)
2. Brinker, K., Fürnkranz, J., Hüllermeier, E.: A Unified Model for Multilabel Classification and Ranking. In: *Proceeding of the 17th European Conference on Artificial Intelligence*. pp. 489–493 (2006)

3. Elisseeff, A., Weston, J.: Kernel methods for Multi-labelled classification and Categorical regression problems. *Advances in Neural Information Processing Systems* 14, 681–687 (2001)
4. Gibaja, E., Ventura, S.: Multi-label learning: a review of the state of the art and ongoing research. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 4(6), 411–444 (2014)
5. Gönen, M., Alpaydm, E.: Multiple kernel learning algorithms. *The Journal of Machine Learning Research* 12, 2211–2268 (2011)
6. He, J., Lawrence, R.: A graph-based framework for multi-task multi-view learning. In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. pp. 25–32 (2011)
7. Jin, X., Zhuang, F., Wang, S., He, Q., Shi, Z.: Shared structure learning for multiple tasks with multiple views. In: *Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science*, vol. 8189, pp. 353–368. Springer Berlin Heidelberg (2013)
8. Manning, C.D., Raghavan, P., Schütze, H.: *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK (2008)
9. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. *Machine Learning* 85(3), 1–27 (2011)
10. Sechidis, K., Tsoumakas, G., Vlahavas, I.: On the stratification of multi-label data. In: *Machine Learning and Knowledge Discovery in Databases*, pp. 145–158. Springer Berlin Heidelberg (2011)
11. Sharma, A., Kumar, A., Daume III, H., Jacobs, D.W.: Generalized multiview analysis: A discriminative latent space. In: *Computer Vision and Pattern Recognition (CVPR)*, 2012 IEEE Conference on. pp. 2160–2167. IEEE (2012)
12. Snoek, C.G., Worring, M., Smeulders, A.W.: Early versus late fusion in semantic video analysis. In: *Proceedings of the 13th annual ACM international conference on Multimedia*. pp. 399–402. ACM (2005)
13. Sun, S.: A survey of multi-view machine learning. *Neural Computing and Applications* 23(7-8), 2031–2038 (2013)
14. Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.: Multi-label Classification of Music into Emotions. *EURASIP Journal on Audio, Speech, and Music Processing* 2011(1), 4 (2011)
15. Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., Vlahavas, I.: Mulan: A Java Library for Multi-Label Learning. *Journal of Machine Learning Research* 12, 2411–2414 (2011)
16. Xu, C., Tao, D., Xu, C.: A survey on multi-view learning. *ArXiv.org (Cornell University)* (2013)
17. Xu, J.: Fast multi-label core vector machine. *Pattern Recognition* 46(3), 885 – 898 (2013)
18. Xu, J.: Laboratory of Intelligent computation. http://computer.njnu.edu.cn/Lab/LABIC/LABIC_Software.html (2013)
19. Zhang, J., Huan, J.: Inductive multi-task learning with multiple view data. In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. pp. 543–551. ACM (2012)
20. Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering* 26(8), 1819–1837 (2014)
21. Zou, F., Liu, Y., Wang, H., Song, J., Shao, J., Zhou, K., Zheng, S.: Multi-view multi-label learning for image annotation. *Multimedia Tools and Applications* pp. 1–18 (2015)