

Clasificación multi-etiqueta con ensamble fijo por mayoría ponderada

Cardona, Juan Cruz; Banchero, Santiago

Universidad Nacional de Luján, Luján, Buenos Aires, Argentina
 juancruzcardona@gmail.com, sbanchero@unlu.edu.ar

Resumen La clasificación multi-etiquetas es un paradigma de aprendizaje supervisado que generaliza las técnicas clásicas de clasificación para abordar problemas en donde cada instancia de una colección se encuentra asociada a múltiples etiquetas. La mayor parte de los trabajos de investigación han sido realizados en contextos de aprendizaje por *batch*. Los ambientes de flujo continuo de datos (o *streaming*) presentan nuevos desafíos a esta área debido a las limitaciones de tiempo de respuesta y almacenamiento que acarrearán. Se aplicaron algoritmos de clasificación multi-etiqueta a diversas colecciones de datos no estructuradas de referencia a partir de las cuales se simuló los *streamings* de datos. En este trabajo se propone una estrategia de ensamble de algoritmos de clasificación multi-etiquetas con el objetivo de conseguir mejoras en la predicción. Los resultados han sido alentadores y la propuesta de ensambles utilizando algoritmos clásicos de clasificación multi-etiquetas mostraron rendimientos competitivos que mejoran en varios escenarios al estado del arte.

Keywords: Multi-Etiquetas · Streaming de datos · Clasificación · Ensamblados · Votación.

1. Introducción

La revolución de los datos en las dos últimas décadas ha cambiado ha generado desafíos importantes en la extracción de conocimiento a partir de los datos. El volumen de datos crece constantemente y estos no pueden ser todos almacenados. La amplia variedad de fuentes generadoras de datos abarcan desde redes de sensores, flujos de *clicks* en la minería web, redes sociales, análisis de *logs*, etc. El análisis en tiempo real de estos flujos de datos se está convirtiendo en un área clave de la investigación en ciencia de datos a medida que aumenta el número de aplicaciones que exigen este tipo de procesamiento.

Un entorno de flujo de datos tiene requisitos diferentes a los del entorno tradicional de aprendizaje tipo *batch*. Los más significativos: 1) procesar un ejemplo a la vez, e inspeccionarlo solo una vez (como máximo), 2) estar preparado para predecir en cualquier momento; 3) los datos pueden evolucionar con el tiempo; y 4) esperar un flujo infinito, pero procesarlo con recursos finitos (tiempo y memoria) [1].

A diferencia del aprendizaje automático tradicional, que usa datos de etiqueta única para representar objetos del mundo real, cada instancia en el aprendizaje multi-etiquetas representa un único objeto pero puede contener más de una etiqueta. Por consiguiente, la tarea de clasificación consiste en hallar una función que logre asignar a cada objeto, nuevo y desconocido, el conjunto de etiquetas que lo caracteriza [6].

Asumiendo que $X = \mathbb{R}^d$ denota el espacio de instancias d dimensional, y que $Y = \{y_1, y_2, \dots, y_q\}$ denota el espacio de etiquetas con q etiquetas posibles, la tarea de clasificación multi-etiquetas consiste en entrenar un conjunto $D = \{(x_i, Y_i) \mid 1 \leq i \leq m\}$ para hallar una función h tal que $h : X \rightarrow 2^Y$. A su vez, X_i es un vector de atributos d dimensional definido como $(x_{i1}, x_{i2}, \dots, x_{id})$. Y_i , por su parte, es el conjunto de etiquetas asociadas a la instancia X_i . Luego, para cada instancia desconocida $x \in X$ el clasificador h predice $h(x) \subseteq Y$ que representa el conjunto de etiquetas hallado para x .

El objetivo de este trabajo es presentar un algoritmo de ensamble fijo por mayoría ponderada (EFMP) utilizando algoritmos de clasificación multi-etiquetas en *streaming* de datos.

2. Datos

Se trabajó con colecciones de datos multi-etiquetas de referencia, utilizadas en la literatura para evaluar la capacidad predictiva de los modelos de clasificación [12,14,2]. La tabla 1 enumera sus características principales, incluyendo métricas que describen su grado de multi-etiquetado.

Tabla 1. Colecciones multi-etiquetas y sus características. N: número de instancias; A: número de atributos; L: número de etiquetas; LC: cardinalidad de etiquetas; LD: densidad de etiquetas.

Nombre	Dominio	N	A	L	LC	LD
20ng	Texto	19300	1006	20	1,029	0,051
Enron	Texto	1702	1001	53	3,378	0,064
Mediamill	Video	43907	120	101	4,376	0,043

20NG es un conjunto de datos que consta de 20 mil publicaciones aproximadamente, provenientes de grupos de noticias y que abordan 20 tópicos diferentes [9]. Enron es una colección de correos electrónicos seleccionados de entre los 500 mil generados por empleados de la compañía eléctrica *Enron* y filtrados - quedando unos 2000 documentos - durante una investigación por corrupción [7]. Finalmente, Mediamill es una colección generada a partir de 80 horas de video provenientes de transmisiones de noticias durante noviembre de 2004 [17]. Se seleccionaron más de 43 mil ejemplos y fue manualmente etiquetada con 101 conceptos.

3. Ensamble fijo por mayoría ponderada

El método de ensamble propuesto utiliza como clasificadores base tres algoritmos de aprendizaje multi-etiqueta diferentes, que se mantienen fijos durante todo el entrenamiento. Además, el ensamble mantiene una ponderación de cada clasificador de acuerdo a su rendimiento, penalizando por cada etiqueta mal clasificada, y la combinación de los votos se lleva a cabo por mayoría de voto ponderada. La implementación se basa en la presentada por [8], quienes también ponderan los clasificadores pero usan un único tipo de clasificador base y no contemplan problemas de múltiples etiquetas. Los experimentos se realizan con dos versiones, una de ellas se entrena con todas las instancias del subconjunto de entrenamientos y la otra tomando muestreos siguiendo la distribución de *Poisson*, tal como se realiza para ensambles del tipo de *Oza Bagging* [13].

Se propone una estrategia de ensamble en ambientes de flujos continuos que pondera a los clasificadores base de acuerdo a su rendimiento y que ajusta los pesos en cada predicción, a fin de optimizar la exactitud y eficiencia de la respuesta, y al mismo tiempo mantenerse actualizado frente a los cambios de concepto. Además, es un ensamble que permite definir clasificadores base modelados a partir de algoritmos de clasificación diferentes para explotar la variabilidad en los mismos. La técnica de ponderación se basa en la presentada por [8] para la estrategia *Dynamic Weighted Majority* (DWM) y fue ajustada para soportar datos de múltiples etiquetas. Otra de las cualidades del ensamble DWM es que agrega y elimina clasificadores base dinámicamente de acuerdo a su ponderación. Sin embargo, el costo computacional acarreado es notorio y los nuevos modelos añadidos son de un mismo tipo y no permite variarlos. En consecuencia, se presenta la estrategia EFMP como posible alternativa junto con una variación del mismo, EFMP2, que muestrea instancias según la distribución *Poisson*. A continuación, se describen ambas técnicas en detalle.

EFMP mantiene un conjunto fijo de m clasificadores base, cada uno con un vector de pesos $W_k = [w_0, w_1, \dots, w_q]$, donde $1 \leq k \leq m$ y $w_{k,j}$ representa el peso del clasificador k para la etiqueta j . En el entrenamiento del modelo se reciben n instancias donde $n = \|D\|$ para la estrategia simple. Además se definen los parámetros p , que es la cantidad de instancias observadas entre actualizaciones de los pesos, y β , que representa el factor en el que se decrece el peso $w_{k,j}$ ante cada clasificación errónea. β es un valor definido en el dominio $0 \leq \beta \leq 1$ y toma el valor 0,5 por defecto. Todos los pesos son inicializados en 1.

El proceso de aprendizaje se lleva a cabo de la siguiente manera: al arribar una instancia i , EFMP se la asigna a cada uno de los m clasificadores. En primer lugar se realiza la actualización de pesos y si un clasificador C_k no predice correctamente una etiqueta j , su peso $w_{k,j}$ será multiplicado por el factor β . Luego, se entrena cada clasificador con la instancia nueva y se repite el procedimiento con la siguiente. El parámetro p es usado durante esta etapa y determina los períodos entre los cuales no se deben actualizar los pesos. Una vez completado un período se normalizan los pesos de manera tal que el máximo peso entre etiquetas es uno. El algoritmo 1 presenta el pseudocódigo de este proceso en detalle.

Algorithm 1: Algoritmo de entrenamiento y ajuste de pesos para EFMP

Input: $\{X, Y\}$: Conjunto de entrenamiento, n : Número de instancias de entrenamiento, q : Número de etiquetas, β : Factor de decrecimiento de los pesos, p : Período entre actualizaciones de los pesos, C : Clasificadores base, W : Pesos de los clasificadores

```

i ← 0
m ← ||C||
while i < n do
    if i mód p = 0 then
        k ← 0
        while k < m do
            yi ← predecir(Ck, Xi)
            j ← 0
            while j < q do
                if yi,j ≠ Yi,j then
                    | Wk,j ← β * Wk,j
                end
                j ← j + 1
            end
            k ← k + 1
        end
        W ← escalarPesos(W)
    end
    k ← 0
    while k < m do
        | entrenar(Ck, Xi, Yi)
        | k ← k + 1
    end
    i ← i + 1
end
    
```

4. Algoritmos

Existe una amplia variedad de algoritmos de clasificación multi-etiquetas con diferentes estrategias de resolución del desafío de tener múltiples posibles rótulos para cada caso del conjunto de datos. Estas estrategias pueden dividirse en 1) Transformación del Problema y 2) Adaptación del Algoritmo. El primero engloba al conjunto de algoritmos que abordan el problema de clasificación multi-etiquetas transformándolo en múltiples problemas de clasificación de única etiqueta, lo cual permite utilizar algoritmos de clasificación convencionales en problemas multi-etiqueta. Tres de estos métodos fueron particularmente relevantes para este trabajo: *Binary Relevance* (BR), *Classifier Chains* (CC) [15]

y *Label Powerset* (Label Powerset) [19]. El segundo abordaje es a partir de la adaptación de algoritmos clásicos y bien conocidos a este tipo de escenarios. La categoría engloba al conjunto de algoritmos que acometen el problema de clasificación multi-etiquetas mediante la modificación de algoritmos de etiqueta única para que sean capaces de manejar la nueva naturaleza de los datos en estas tareas. En lo que confiere a ambientes de flujos continuos de datos una de las técnicas más populares es *Hoeffding Tree* (HT) [3]. HT tiene su adaptación a problemas de multi-etiquetas (MLHT) [14] utilizando una versión optimizada de LP llamada *Label Combination* (LC).

Los experimentos fueron corridos utilizando los algoritmos multi-etiquetas disponibles en la librería *scikit-multiflow* [11] y la implementación de ensamble presentada en este trabajo ¹ EFMP y su variación EFMP2. Los algoritmos BR y CC usan *Naïve Bayes* como modelo de clasificación base y MLHT es ejecutado en su versión basada en Label Powerset [14].

En lo que respecta a soluciones de ensamble, los modelos de EFMP fueron configurados en ambos casos con tres clasificadores base: CC, BR y MLHT. La comparación se hará contra el algoritmo DWM [8] pero adaptado a ambientes de multi-etiquetas y se suman al análisis los algoritmos de *Ensamble de Binary Relevance* (EBR) y *Ensamble de Classifier Chains* (ECC) [13]. Los tres algoritmos de ensamble extraídos de la literatura son configurados con diez clasificadores base de *Naïve Bayes* [12,14,2].

5. Evaluación

La evaluación de los resultados fue realizada con un conjunto de métricas ampliamente utilizadas en el estado del arte, tanto en escenarios de flujos [18,21,12] como en *batch* [10,20,5]. En este sentido se incluyeron dos estrategias de evaluación de rendimiento de los modelos con F-Score (F1) tanto para una evaluación basada en ejemplos como para basadas en etiquetas. Y una tercera evaluación orientada a medir la eficiencia tanto en tiempo de cómputo como tamaño de los modelos obtenidos.

Para la etapa de evaluación se aplicó la técnica de evaluación secuencial predictiva (*prequential*) con ventanas deslizantes, tal como se recomienda para ambientes de flujos continuos [4]. Ante cada ejemplo - o ventana de ejemplos arribada - el modelo primero realiza la predicción y luego el entrenamiento. Finalmente las métricas de evaluación son calculadas una vez procesados todos los ejemplos de la colección y a partir de todas las predicciones producidas. A partir de esta técnica el modelo predice y entrena todas las instancias, y no solo un subconjunto de ellas como sucede con la estrategia de *holdout*. La ventana deslizante se configura en $w = \frac{N}{20}$, es decir, se divide el número total de instancias del flujo en 20 ventanas, siguiendo las directivas de [14]. Los resultados de la evaluación son agrupados según los tipos de métrica usados para facilitar el análisis.

¹ <https://github.com/Juancard/multi-label-classification-on-data-streamings>

6. Resultados

La metodología propuesta permitió evaluar los diferentes algoritmos de clasificación multi-etiqueta para los diferentes *streams* utilizando las configuraciones sin ensambles, los ensambles de referencias y los métodos de ensamble propuestos. Los resultados se dividen en métricas de ajustes del modelo F-Score (F1) basadas en ejemplos (Tabla 2), métricas basadas en etiquetas (tablas 3 y 4) y por último las métricas de eficiencia (tabla 5) que cuantifican el tiempo de procesamiento y espacio de almacenamiento de los modelos.

Los valores de F1 obtenidos para la evaluación basada en ejemplos (Tabla 2) muestra que EFMP y EFMP2 fueron mejores que los *baselines* de ensambles en todos los casos. Además, superó a los que no utilizan ensambles en el *stream* de Enron. En los casos de 20NG, EFMP fue superado en un 3% por BR y en Mediamill MLHT superó a EFMP en un 3.5%.

Tabla 2. Valores de F-Score (F1) basada en ejemplos de *stream* 20NG, Enron y Mediamill para los distintos algoritmos evaluados.

<i>Stream</i>	F1 Basada en ejemplos		
	20NG	Enron	Mediamill
Binary Relevance (<i>Naïve</i> Bayes)	0.329	0.345	0.163
Classifier Chains (<i>Naïve</i> Bayes)	0.315	0.344	0.311
Multi-label Hoeffding Tree	0.324	0.172	0.426
DWM+BR (<i>Naïve</i> Bayes)	0.192	0.229	0.214
DWM+CC (<i>Naïve</i> Bayes)	0.181	0.231	0.400
Ensamble de Binary Relevance (<i>Naïve</i> Bayes)	0.198	0.292	0.171
Ensamble de Classifier Chains (<i>Naïve</i> Bayes)	0.142	0.283	0.388
EFMP2 (BR, CC y MLHT)	0.264	0.340	0.382
EFMP (BR, CC y MLHT)	0.328	0.365	0.411

En la comparación basada en etiquetas, Macro-F1 (tabla 3) permitió ver el F1 calculado por clases y nos muestra el valor promedio de la métrica. Aquí se observa que en promedio EFMP es el mejor en el *stream* de 20NG mientras que EFM2 lo es para Mediamill. En el caso de Enron el mejor es el algoritmo BR que supera en un 11% a EFMP. También se observó un mejor rendimiento de las propuestas de ensamble EFMP y EFMP2 con relación a los ensambles de referencia.

Micro-F1 nos permite comparar el rendimiento de los algoritmos a partir de la sumatoria individual de verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos. En este caso EFMP es el mejor algoritmo tanto para 20NG como para Enron y el algoritmo MLHT es el mejor para Mediamill. Este último *stream* es el único caso donde además hay ensambles del *baseline* que son mejores que los modelos propuestos, ECC supera a EFMP en un 0.5% (tabla 4). En el resto de los *streams* (20NG y Enron) los algoritmos EFMP y EFMP2 son mejores que los otros ensambles.

Clasificación multi-etiqueta con ensamble fijo por mayoría ponderada 7

Tabla 3. Valores de Macro F-Score (F1) de *stream* 20NG, Enron y Mediamill para los distintos algoritmos evaluados.

<i>Stream</i>	Macro F1		
	20NG	Enron	Mediamill
Binary Relevance (<i>Naïve</i> Bayes)	0.461	0.108	0.111
Classifier Chains (<i>Naïve</i> Bayes)	0.450	0.105	0.091
Multi-label Hoeffding Tree	0.402	0.008	0.043
DWM+BR (<i>Naïve</i> Bayes)	0.314	0.049	0.111
DWM+CC (<i>Naïve</i> Bayes)	0.299	0.047	0.084
Ensamble de Binary Relevance (<i>Naïve</i> Bayes)	0.326	0.081	0.111
Ensamble de Classifier Chains (<i>Naïve</i> Bayes)	0.250	0.067	0.068
EFMP2 (BR, CC y MLHT)	0.396	0.083	0.124
EFMP (BR, CC y MLHT)	0.464	0.096	0.073

Tabla 4. Valores de Micro F-Score (F1) de *stream* 20NG, Enron y Mediamill para los distintos algoritmos evaluados.

<i>Stream</i>	Micro F1		
	20NG	Enron	Mediamill
Binary Relevance (<i>Naïve</i> Bayes)	0.445	0.347	0.168
Classifier Chains (<i>Naïve</i> Bayes)	0.433	0.358	0.306
Multi-label Hoeffding Tree	0.331	0.124	0.410
DWM+BR (<i>Naïve</i> Bayes)	0.313	0.269	0.218
DWM+CC (<i>Naïve</i> Bayes)	0.298	0.272	0.384
Ensamble de Binary Relevance (<i>Naïve</i> Bayes)	0.320	0.337	0.175
Ensamble de Classifier Chains (<i>Naïve</i> Bayes)	0.243	0.333	0.395
EFMP2 (BR, CC, MLHT)	0.387	0.369	0.382
EFMP (BR, CC, MLHT)	0.447	0.374	0.393

Los resultados de eficiencia mostraron (tabla 5) que los algoritmos de ensamble propuestos (EFMP y EFMP2) reducen tanto el espacio de almacenamiento como el tiempo de procesamiento de los *streams*. Las métricas de tiempo se reducen significativamente para 20NG y Enron, mientras que en Mediamill EBR hace un uso significativamente menor de tiempo que los ensambles presentados.

El resultado de los métodos propuestos muestra que son competitivos respecto a la literatura de referencia. En particular, para las pruebas realizadas con el conjunto de datos 20ng los valores de F-Score basado en ejemplos obtenidos superan en 2.3% a [12] pero no son mejores que otros como [18], [2] y [16]. En las pruebas realizadas con el Enron, los métodos propuestos superan en 10.74% a [12], duplican el rendimiento de [16] y son superados por [18] en un 22.6%. Finalmente, para el conjunto de datos Mediamill tanto [18] como [16] superan nuestra propuesta.

Tabla 5. Valores de eficiencia de *streams* 20NG, Enron y Mediamill para los distintos algoritmos evaluados.

<i>Stream</i>	Tamaño (KiB)		Tiempo			
	20NG	Enron	Mediamill	20NG	Enron	Mediamill
Binary Relevance (<i>Naïve</i> Bayes)	31.6	82.9	18.9	1:43:26	0:21:28	2:08:42
Classifier Chains (<i>Naïve</i> Bayes)	31.9	85.1	26.7	1:45:00	0:27:05	2:52:36
Multi-label Hoeffding Tree	22.8	284.7	306.0	2:22:30	0:58:16	19:05:49
DWM+BR (<i>Naïve</i> Bayes)	305.5	782.9	185.3	21:23:37	5:15:03	1 día, 4:21:27
DWM+CC (<i>Naïve</i> Bayes)	308.4	802.7	262.4	21:03:02	5:05:56	1 día, 12:49:14
EBR (<i>Naïve</i> Bayes)	316.1	826.8	189.2	14:14:56	3:41:00	20:13:01
ECC (<i>Naïve</i> Bayes)	319.0	847.6	267.2	14:35:33	4:03:16	1 día, 1:23:01
EFMP2 (BR, CC, MLHT)	84.7	359.2	262.2	5:36:55	2:03:59	1 día, 6:27:51
EFMP (BR, CC, MLHT)	86.4	452.8	351.7	5:41:58	1:45:51	1 día, 0:47:17

7. Conclusiones

La clasificación multi-etiquetas representa un gran desafío que se incrementa en el contexto de los *streamings* de datos. Dada esta complejidad todas las mejoras en el rendimiento de estos algoritmos de clasificación son un aporte muy valioso en vías de incrementar la capacidad de predicción.

La utilización de ensambles en clasificaciones multi-etiquetas produjo mejores métricas de evaluación para los algoritmos de ensamble fijo por mayoría ponderada y su variante. En la evaluación por etiquetas Micro y Macro F1 - la mayoría de los casos son mejores que los ensambles de referencia. Estos no superan nunca a los ensambles EFMP y EFMP2 en ninguna de las métricas a excepción de ECC en Micro-F1. El análisis basado en ejemplos favoreció a los métodos que no utilizan ensambles para 20NG y para Mediamill. Mientras que en Enron el mejor algoritmo fue EFMP.

8. Agradecimientos

Al Centro de Investigación, Docencia y Extensión en Tecnologías de la Información y las Comunicaciones (CIDETIC) de la Universidad Nacional de Luján.

Referencias

1. Bifet, A., Gavaldà, R.: Adaptive Learning from Evolving Data Streams. pp. 249–260 (Aug 2009). https://doi.org/10.1007/978-3-642-03915-7_22
2. Büyükçakır, A., Bonab, H., Can, F.: A Novel Online Stacked Ensemble for Multi-Label Stream Classification. Proceedings of the 27th ACM International Conference on Information and Knowledge Management pp. 1063–1072 (Oct 2018). <https://doi.org/10.1145/3269206.3271774>, <http://arxiv.org/abs/1809.09994>, arXiv: 1809.09994
3. Domingos, P., Hulten, G.: Mining High-Speed Data Streams. Proceeding of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Nov 2002). <https://doi.org/10.1145/347090.347107>

4. Gama, J., Sebastião, R., Rodrigues, P.P.: On evaluating stream learning algorithms. *Machine Learning* **90**(3), 317–346 (Mar 2013). <https://doi.org/10.1007/s10994-012-5320-9>, <https://doi.org/10.1007/s10994-012-5320-9>
5. Gibaja, E., Ventura, S.: A Tutorial on Multi-Label Learning. *ACM Computing Surveys* **47** (Apr 2015)
6. Godbole, S., Sarawagi, S.: Discriminative Methods for Multi-Labeled Classification. vol. vol. 3056 (Aug 2004). https://doi.org/10.1007/978-3-540-24775-3_5
7. Klimt, B., Yang, Y.: The Enron Corpus: A New Dataset for Email Classification Research. In: Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M.Y., Weikum, G., Boulicaut, J.F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) *Machine Learning: ECML 2004*, vol. 3201, pp. 217–226. Springer Berlin Heidelberg, Berlin, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30115-8_22, http://link.springer.com/10.1007/978-3-540-30115-8_22, series Title: Lecture Notes in Computer Science
8. Kolter, J.Z., Maloof, M.A.: Dynamic Weighted Majority: An Ensemble Method for Drifting Concepts. *The Journal of Machine Learning Research* **8**, 2755–2790 (Dec 2007)
9. Lang, K.: NewsWeeder: Learning to Filter Netnews. In: Prieditis, A., Russell, S. (eds.) *Machine Learning Proceedings 1995*, pp. 331–339. Morgan Kaufmann, San Francisco (CA) (Jan 1995). <https://doi.org/10.1016/B978-1-55860-377-6.50048-7>, <http://www.sciencedirect.com/science/article/pii/B9781558603776500487>
10. Madjarov, G., Kocev, D., Gjorgjevikj, D., Džeroski, S.: An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition* **45**(9), 3084–3104 (Sep 2012). <https://doi.org/10.1016/j.patcog.2012.03.004>, <https://linkinghub.elsevier.com/retrieve/pii/S0031320312001203>
11. Montiel, J., Read, J., Bifet, A., Abdessalem, T.: Scikit-Multiflow: A Multi-output Streaming Framework. arXiv:1807.04662 [cs, stat] (Jul 2018). <https://doi.org/10.5555/3291125.3309634>, <http://arxiv.org/abs/1807.04662>, arXiv: 1807.04662
12. Osojnik, A., Panov, P., Džeroski, S.: Multi-label classification via multi-target regression on data streams. *Machine Learning* **106**(6), 745–770 (Jun 2017). <https://doi.org/10.1007/s10994-016-5613-5>, <https://doi.org/10.1007/s10994-016-5613-5>
13. Oza, N.C.: Online bagging and boosting. In: 2005 IEEE International Conference on Systems, Man and Cybernetics. vol. 3, pp. 2340–2345 Vol. 3 (Oct 2005). <https://doi.org/10.1109/ICSMC.2005.1571498>, ISSN: 1062-922X
14. Read, J., Bifet, A., Holmes, G., Pfahringer, B.: Scalable and efficient multi-label classification for evolving data streams. *Machine Learning* **88**(1), 243–272 (Jul 2012). <https://doi.org/10.1007/s10994-012-5279-6>, <https://doi.org/10.1007/s10994-012-5279-6>
15. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. *Mach. Learn.* **85**(3), 333–359 (2011)
16. Roseberry, M., Cano, A.: Multi-label kNN Classifier with Self Adjusting Memory for Drifting Data Streams. In: Second International Workshop on Learning with Imbalanced Domains: Theory and Applications. pp. 23–37. PMLR (Nov 2018), <http://proceedings.mlr.press/v94/roseberry18a.html>, ISSN: 2640-3498

- 10 JC. Cardona et al.
17. Snoek, C.G.M., Worring, M., van Gemert, J.C., Geusebroek, J.M., Smeulders, A.W.M.: The challenge problem for automated detection of 101 semantic concepts in multimedia. In: Proceedings of the 14th annual ACM international conference on Multimedia - MULTIMEDIA '06. p. 421. ACM Press, Santa Barbara, CA, USA (2006). <https://doi.org/10.1145/1180639.1180727>, <http://portal.acm.org/citation.cfm?doid=1180639.1180727>
 18. Sousa, R., Gama, J.: Multi-label classification from high-speed data streams with adaptive model rules and random rules. *Progress in Artificial Intelligence* (2018)
 19. Tsoumakas, G., Katakis, I., Vlahavas, I.: Random k-Labelsets for Multi-Label Classification. *IEEE Trans. Knowl. Data Eng.* **23**, 1079–1089 (Jul 2011). <https://doi.org/10.1109/TKDE.2010.164>
 20. Zhang, M.L., Zhang, K.: Multi-label learning by exploiting label dependency. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '10 (2010)
 21. Zheng, X., Li, P., Chu, Z., Hu, X.: A Survey on Multi-Label Data Stream Classification. *IEEE Access* **8**, 1249–1275 (2020). <https://doi.org/10.1109/ACCESS.2019.2962059>, conference Name: IEEE Access