Estudio Comparativo de Técnicas de Balanceo de Datos en el Aprendizaje de Múltiples Instancias

Jose Arrieta*, Carlos Mera[†]

*Universidad Nacional de Colombia – Sede Medellín, Medellín, Colombia
[†]Universidad de Medellín, Medellín, Colombia
jmarrietar@unal.edu.co – cmera@udem.edu.co

Palabras Clave: Aprendizaje de múltiples instancias, Desbalanceo de clases, Sobre-muestreo, Sub-muestreo.

Resumen

En el aprendizaje supervisado se considera que un conjunto de entrenamiento de dos clases está desbalanceado cuando el número de muestras de una de las clases (la clase *mayoritaria*) sobrepasa el número de muestras de la otra (la clase *minoritaria*). Diferentes estudios han mostrado que el desempeño de la mayoría de algoritmos de clasificación, basados en la teoría de decisión de Bayes, se afecta negativamente cuando estos son entrenados con conjuntos de datos desbalanceados. A pesar de que este problema ha sido ampliamente estudiado en el aprendizaje de una sola instancia (SIL), poca atención se ha prestado a él en el contexto del aprendizaje de múltiples instancias (MIL). Con base en lo anterior, en este trabajo se discute el problema del aprendizaje con clases desbalanceadas en el contexto MIL y se hace un estudio comparativo de algunas de las técnicas clásicas de muestreo para resolver este problema en MIL. La evaluación de los métodos se hace sobre once conjuntos de datos de referencia que presentan diferentes niveles de desbalance entre las clases. Los resultados experimentales muestran que la aplicación directa de estos métodos en conjuntos de datos tipo MIL no es pertinente.

1 Introducción

En los últimos años, un paradigma de clasificación denominado Aprendizaje de Múltiples Instancias (MIL) [1] ha venido ganando terreno entre las aplicaciones de la visión por computador. Esto se debe a la capacidad de este nuevo paradigma de tratar con conjuntos de datos débilmente etiquetados, en los que se asume que hay cierta ambigüedad en la forma como se asignan las etiquetas. En MIL, un objeto, llamado bolsa, es representado por múltiples vectores de características, o instancias, las cuales a menudo se corresponden con los segmentos de los objetos en las imágenes [2]. Cada bolsa tiene asociada una etiqueta de clase, sin embargo, las etiquetas de las instancias dentro de las bolsas se asumen desconocidas.

En un problema de clasificación de dos clases, el supuesto estándar en MIL, es que una bolsa positiva contiene al menos una instancia positiva (el concepto objetivo), mientras que una bolsa negativa solo contiene instancias negativas [3]. Esto significa que no todas las instancias en las bolsas positivas son necesariamente relevantes y, en consecuencia, éstas pueden tener instancias negativas que causan ambigüedad en las mismas.

Un problema relacionado con el aprendizaje de múltiples instancias, en el contexto de la visión por computador, es el desbalanceo de clases. Este problema hace referencia al entrenamiento de un clasificador MIL en el que el número de muestras de una clase (la clase mayoritaria) supera en mucho el número de muestras de la otra (la clase minoritaria). Por ejemplo, en el área de la inspección visual automática, es común tener muchas muestras de objetos sin defectos (la clase mayoritaria) y muy pocas muestras de objetos con defectos (la clase minoritaria); el mismo problema se presenta en aplicaciones de reconocimiento de rostros en video-vigilancia donde suelen haber pocos ejemplos del objetivo (la clase minoritaria) comparados con el fondo de la escena (la clase mayoritaria).

En la literatura existen diferentes aproximaciones para tratar con el problema de clases desbalanceadas en el contexto del aprendizaje de una sola instancia [4]–[6] . Sin embargo, en MIL la mayoría de los algoritmos no consideran este problema el cual afecta negativamente el desempeño en la clasificación. Con base en lo anterior, en este trabajo se discute el problema de clases desbalanceadas en el aprendizaje de múltiples instancias y se hace un estudio comparativo de algunas de las técnicas de balanceo, propuestas en el aprendizaje de una sola instancia, aplicadas a conjuntos de datos tipo MIL. La estructura de este trabajo es la siguiente. En la Sección 2 se introduce el problema de desbalanceo de clases y se presentan algunos de los métodos comúnmente utilizados para solucionar el mismo. En la Sección 3 se hace una revisión del aprendizaje de múltiples instancias y se introduce el problema de desbalanceo de clases en MIL. En la Sección 4 se describen los experimentos realizados y se discuten los resultados obtenidos. En la Sección 5 se presentan las conclusiones del estudio y se direccionan los posibles trabajos futuros.

2 El Problema de Clases Desbalanceadas

En el aprendizaje supervisado, el problema de la representación desigual de clases, también conocido como el problema de clases desbalanceadas, se presenta cuando en el conjunto de datos de entrenamiento no hay un número (aproximadamente) igual de muestras de cada clase [7]. Este problema resulta particularmente importante en aquellos dominios de aplicación en los que clasificar erróneamente un objeto de la clase minoritaria tiene un costo medio muy elevado.

Los algoritmos de aprendizaje basados en la teoría de decisión Bayesiana, buscan la regla decisión óptima que minimice el riesgo global en la clasificación. Cuando esta búsqueda se hace con base en un conjunto de entrenamiento con clases altamente desbalanceadas, la regla de decisión produce fronteras de decisión sesgadas en favor de la clase mayoritaria. En consecuencia, los objetos en la clase minoritaria, y generalmente la más importante, tienden a ser erróneamente clasificados [8].

Durante los últimos años se han realizado muchos esfuerzos para proporcionar soluciones a este problema. Por ejemplo, se han desarrollado algoritmos de muestreo para cambiar las distribuciones de las clases [9]–[11]; se han propuesto técnicas de aprendizaje basadas en costos que castigan la clasificación errónea de los objetos en la clase minoritaria [12], [13]; se han utilizado técnicas de clasificación de una clase para modelar solo los objetos de clase minoritaria [14], [15]; y se han usado técnicas ensambles de clasificadores con las que se busca centrar la atención de cada miembro del ensamble en los objetos de la clase minoritaria [16], [17].

De entre éstos, los métodos de muestro tienen ciertas ventajas sobre los demás: son independientes del algoritmo de clasificación que se use; tienen un costo computacional relativamente bajo en la mayoría de los casos de aplicación; y su utilización mejora el desempeño de la clasificación en la mayoría de conjuntos de prueba con clases desbalanceadas [18]. No obstante, también tienen algunas desventajas: no está claro cuál es el método de muestreo que obtiene los mejores resultados y su desempeño siempre está relacionado con el dominio de aplicación, la proporción de datos que se agregan o eliminan del conjunto original y con las medidas de desempeño que se usen para su evaluación [19].

Los métodos de muestreo más simples son el sobre-muestreo y el sub-muestro aleatorios. En el primero se seleccionan al azar muestras de la clase minoritaria, las cuales se duplican y se agregan al conjunto de datos original, mientras que en el segundo se eliminan muestras al azar escogidas de la clase mayoritaria. Estas estrategias tienen algunas desventajas: el sobre-muestreo aleatorio puede llevar a problemas de sobreajuste del clasificador, mientras que el sub-muestreo aleatorio puede eliminar información importante para la definición de las fronteras de decisión [20].

Otras estrategias consisten en hacer un muestreo informativo en el que se agregan o eliminan muestras de forma más inteligente. Entre los métodos representativos de sobre-muestreo informativo están: SMOTE (*Synthetic Minority Over Sampling Technique*) [10], el cual crea datos sintéticos a partir de los segmentos de línea que unen dos muestras de la clase minoritaria; Borderline-SMOTE [21], que es una variación de SMOTE que enfoca su atención en la generación de muestras de la clase minoritaria en la frontera de decisión; y ADASYN (*Adaptitve Synthetic Sampling Approach*) [11], que es otra variante de SMOTE que se centra en generar un mayor número de datos sintéticos a partir de las muestras que son consideradas "peligrosas" de acuerdo a la clase a la que pertenecen sus vecinos.

Por otro lado, están los métodos de sub-muestreo informativo los cuales intentan eliminar de la clase mayoritaria aquellas muestras que son consideradas redundantes. Algunos métodos en este grupo son: *Tomek-Links* [22], el cual intenta disminuir el traslape entre clases buscando las muestras cuyos vecinos más cercanos pertenecen a la clase contraria y eliminando entre estos los que corresponden a la clase mayoritaria; CNNR (*Condensed Nearest Neighbor Rule*) [23], el cual tiene como objetivo limpiar los puntos de las zonas distantes a la frontera de decisión; y OSS (*One Side Selection*) [24], el cual combina los dos métodos anteriores. Otros métodos de muestreo más complejos consisten en, por ejemplo, la utilización de algoritmos de *clustering* para seleccionar las muestras a partir de las cuales hacer el sub-muestreo en la clase mayoritaria [25] o incluso el uso de algoritmos bio-inspirados [26] para el mismo fin.

El problema del desbalanceo de clases ha sido ampliamente estudiado por la comunidad científica y numerosas soluciones han sido diseñadas para ser usadas en los métodos de clasificación de una sola instancia. Sin embargo, en el contexto del aprendizaje de múltiples instancias este problema ha sido poco discutido a pesar de ser más complejo.

3 El Aprendizaje de Múltiples Instancias y el Problema de Clases Desbalanceadas

El aprendizaje de múltiples instancias surge del trabajo de Dietterich *et al.* [27] a partir del problema de clasificación de moléculas polimorfas. Para solucionar dicho problema, los autores propusieron una configuración en la que, a diferencia del aprendizaje de supervisado estándar, un objeto es representado por una colección de instancias, o vectores de características, agrupados en bolsas. En este sentido, en MIL el proceso de aprendizaje se realiza con base en las bolsas y no con base en las instancias individuales.

En el caso específico de un problema de dos clases, un conjunto MIL toma la forma $B = \{(B_1, y_1), \dots, (B_n, y_n)\}$, donde, cada bolsa $B_i = \{x_{i1}, \dots, x_{in_i}\}$ es una colección de n_i instancias, y tiene etiqueta $y_i = +1$, para la clase positiva, o $y_i = -1$, para la clase negativa. A partir del supuesto estándar en MIL [3], la etiqueta de una bolsa (y_i) se define a partir de las etiquetas de las instancias (y_{ij}), las cuales se asume que existen, pero son desconocidas. De esta manera,

una bolsa es positiva si esta contiene al menos una instancia positiva (con etiqueta $+1$) o negativa si todas las instancias que esta contiene son negativas (con etiqueta -1).

En su trabajo, Jaume Amores [1] propone una taxonomía donde los algoritmos para MIL están agrupados en tres paradigmas. En el primero, llamado *Instance Space* (IS), el proceso de aprendizaje se realiza al nivel de las instancias, es decir, los clasificadores en él se entrenan para separar las instancias de las bolsas positivas de las instancias en las bolsas negativas. APR (*Axis-Parallel Rectangles*) [27] y mi_SVM [28], son dos algoritmos MIL clásicos en la literatura que caen dentro de este paradigma. La idea de APR es construir un hiper-rectángulo, en el espacio de características, tal que éste solo contenga instancias de las bolsas positivas y ninguna instancia dentro de las bolsas negativas. Por otro lado, mi_SVM, modifica las restricciones en el proceso de optimización de una SVM clásica para incluir el supuesto estándar MIL. MIL-Boost [29] es otro algoritmo en este paradigma cuya base es *AdaBoost*. El segundo paradigma propuesto por Amores es *Bag Space (BS)*. En este paradigma cada bolsa es tratada como una entidad completa y, por tanto, el proceso de discriminativo se realiza entre las bolsas. Un algoritmo en este paradigma es una variante del algoritmo *k*-NN denominada Citation *K*-NN la cual usa una distancia tipo *Hausdorff* para determinar la vecindad entre bolsas. Los algoritmos en el último paradigma, denominado *Embedded Space (ES)*, mapean cada bolsa a un vector de características que condensa la información relevante de la misma. En consecuencia, el proceso discriminativo se realiza en un espacio embebido en el que cada bolsa es representada por un punto en dicho espacio. Dos algoritmos en el paradigma *ES* son MILES [30] y MILIS [31], los cuales implementan una función de mapeo basada en las distancias entre un conjunto de instancias prototipos y las instancias de cada bolsa.

Ninguno de los algoritmos MIL ha considerado, directamente, el problema de la representación desigual de las clases, a pesar de que este es un problema intrínseco en la mayoría de conjuntos de datos de este tipo. Desde el punto de vista MIL, el desbalance entre clases es más complejo ya que este se puede presentar tanto al nivel de las instancias, como al nivel de las bolsas, o en ambos, como se ilustra en la Fig. 1. A primera vista, el desbalance al nivel de instancias se presenta cuando, el número de instancias en las bolsas positivas es mucho menor que el número de instancias en las bolsas negativas (Fig. 1a). Sin embargo, a pesar de que el número de instancias entre bolsas positivas y negativas sea similar, puede aún existir un desequilibrio si el número de instancias que representan el concepto objetivo (instancias realmente positivas en bolsas positivas) es mucho menor que el número de instancias negativas entre las bolsas positivas y negativas. Por otro lado, El desbalance al nivel de bolsas se presenta cuando el número de bolsas positivas es mucho menor que el número de bolsas negativas (Fig. 1b). Finalmente, se considera un desbalance al nivel de instancias y bolsas cuando se presentan los dos casos anteriores de manera simultánea, como se muestra en la Fig. 1c.

En la actualidad este problema ha sido poco explorado en el contexto de MIL, y esa es la razón por la cual en la literatura solo hay un par de trabajos relacionados. En [32] los autores propone dos aproximaciones para tratar con este problema. La primera consiste en un algoritmo de sobre-muestreo de bolsas e instancias basado en SMOTE, mientras que la segunda aproximación es una variante de *AdaBoost* que involucra costos en el proceso de aprendizaje. Por otro lado, en [33], los autores proponen encontrar las instancias más positivas, en las bolsas positivas, y las instancias más negativas, tanto en las bolsas positivas como negativas, y a partir de ellas sobre-muestrear las instancias positivas en las bolsas positivas y sub-muestrear las instancias negativas en la frontera de decisión.

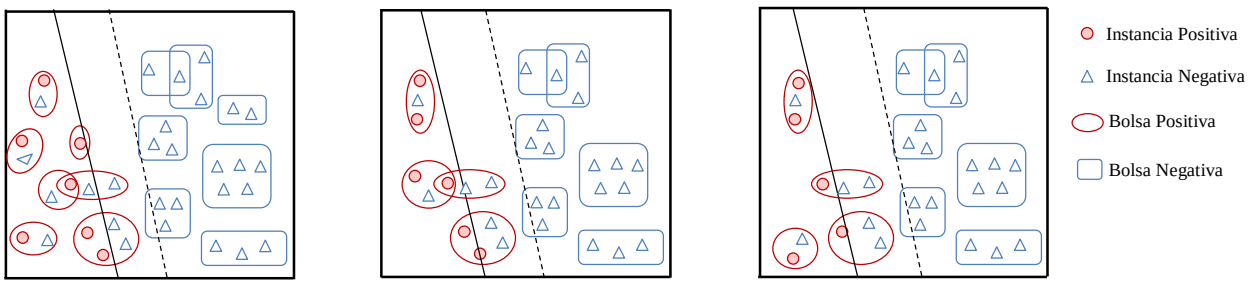
Con base en lo anterior, en este trabajo se evalúa la aplicación directa de algunos de los métodos clásicos de muestreo a conjuntos de datos tipo MIL, con el fin de abordar el problema de desbalanceo al nivel de instancias en conjuntos de datos tipo MIL.

4 Experimentos y Discusión de Resultados

4.1 Conjuntos de Datos

Para este estudio se usaron once conjuntos de datos MIL, provenientes de diferentes dominios de aplicación y con diferentes niveles de desequilibrio entre clases. La Tabla 1 lista estos conjuntos de datos y sus características.

Musk1 y *Musk2* son las bases de datos más populares en MIL dado que provienen de los primeros trabajos de Dietterich *et al.* [27]. La tarea principal en estos conjuntos de



(a) Desbalanceo de Instancias (b) Desbalanceo de Bolsas (c) Desbalanceo de Instancias y Bolsas

Figura 1. Desbalanceo en Conjuntos de Datos MIL. La línea punteada indica la frontera de decisión ideal mientras que la línea continua indica la frontera de decisión generada por el clasificador, claramente sesgada en favor de la clase mayoritaria.

Conjuntos de Datos	Bolsas +	Bolsas -	Inst +	Inst -	Pro	Min	Max
Musk1	47	45	207	269	5	2	40
Musk2	39	63	1017	5581	65	1	1044
Elephant	100	100	762	629	7	2	13
Fox	100	100	647	673	7	2	13
Tiger	100	100	544	676	6	1	13
Muta1	125	63	7790	2696	56	28	88
Muta2	13	29	660	1472	51	26	86
Bird WIWR	109	439	1824	8408	19	2	43
Bird BRCR	197	351	4759	5473	19	2	43
Web1	17	58	488	1724	29	4	131
Web2	18	57	499	1720	30	5	200

Tabla 1. Detalles de los conjuntos de datos tipo MIL usados en los experimentos[1]: número de bolsas positivas (+) y negativas (-), número de instancias (Inst) positivas y negativas, número promedio (Pro), máximo (Max) y mínimo (Min) de instancias por bolsa.

datos es clasificar moléculas como "*musk*" y "*non-musk*". *Elephant*, *Fox* y *Tiger* provienen de la base de datos de imágenes de Corel y fueron introducidas en [31]. La tarea en estos conjuntos de datos es clasificar imágenes en una de 20 categorías predefinidas. En este trabajo se usan solo las 3 mencionadas. Mutagenesis (*Muta1* y *Muta2*), son dos conjuntos de datos provenientes del problema de predicción de la actividad en medicamentos [34]. *Birds* es una base de datos obtenida de la grabación diferentes especies de aves. En este trabajo se usaron las grabaciones de dos de las especies: *Brown Creeper* (BRCR) y *Winter Wren* (WIWR). Finalmente, Web1 y Web2 [35] son dos bases de datos provenientes del problema de recomendación de páginas web y la tarea consiste en clasificar una página web como interesante o no.

4.2 Estrategia de Comparación

En la mayoría de bases de datos tipo MIL, el número de instancias entre bolsa y bolsa es diferente, por tanto estos están desbalanceados al nivel de instancias. En este sentido, en este trabajo se compara el desempeño de una estrategia de sobre-muestreo, dos de sub-muestreo y una estrategia mixta, con el fin de determinar la viabilidad de aplicar los métodos de muestreo en conjuntos de datos tipo MIL. Estas estrategias se describen a continuación:

- La estrategia de sobre-muestreo utilizada se basada en SMOTE y consiste en agregar a cada bolsa el número de instancias sintéticas necesarias para que todas las bolsas tengan el número máximo de instancias. Las instancias usadas como punto de referencia por SMOTE son escogidas aleatoriamente dentro de cada bolsa.
- La primera estrategia de sub-muestreo consiste en eliminar de cada bolsa tantas instancias al azar, como sea necesario para que cada bolsa tenga el número mínimo de instancias.
- La segunda estrategia de sub-muestreo consiste en aplicar el algoritmo OSS al conjunto de datos replicando la etiqueta de cada bolsa a todas sus instancias.
- Finalmente, la estrategia mixta consiste usar SMOTE para generar instancias sintéticas para aquellas bolsas que tengan menos del número promedio de instancias y usar un sub-muestreo aleatorio para aquellas bolsas que tengan más del número promedio de instancias.

4.3 Resultados y Discusión

Adoptando un procedimiento estándar para la evaluación, los experimentos se repitieron 10 veces usando conjuntos de entrenamiento y prueba aleatorios en una validación cruzada con 5 particiones, es decir, en total se realizaron 50 pruebas por cada conjunto de datos. Para medir el desempeño de los algoritmos se usaron la métrica F1 y área bajo la curva (AUC), las cuales son medidas de desempeño estándar para problemas de clases desbalanceadas [5]. Adicionalmente, Se probaron diferentes algoritmos de clasificación MIL reportados en la literatura, sin embargo, solo se presentan los resultados con aquellos que obtuvieron el mejor desempeño de clasificación con el conjunto de datos original, es decir, antes de aplicar las estrategias de muestreo. Los algoritmos usados se listan abajo junto con sus parámetros.

- APR con un umbral $t = 0{,}1$
- Citation *k*-NN con *k*=3 (denominado C-kNN)
- mi-SVM con un kernel de base radial y un parámetro de regularización $C = 10$.
- MILES con un kernel de base radial y un parámetro de regularización $C = 10$.

Los resultados obtenidos se resumen en las Tabla 2 y 3, para las métricas AUC y F1, respectivamente. En ambas tablas bajo el nombre de cada conjunto de datos está en corchetes el algoritmo MIL usado con ese conjunto de datos. Las columnas reportan el desempeño obtenido con el conjunto de datos original y las estrategias de muestro implementadas: una estrategia de sobre-muestreo, dos estrategias de sub-muestreo y la estrategia mixta. Los mejores resultados, entre las estrategias de muestreo, son resaltados en negrita para cada conjunto de datos.

En los conjuntos de datos *Elephant* y *Muta1* no se aplicó el método OSS debido a que el número de instancias en las bolsas positivas (la clase de interés) es mayor al número instancias en las bolsas negativas, por tanto no tiene sentido hacer un sub-muestreo a las bolsas negativas. Por otro lado, al utilizar el método de sobre-muestreo sobre el conjunto de datos *musk2* se generaron una gran cantidad de instancias lo que impidió realizar la prueba por requerimientos de memoria.

Conjuntos de datos	Original	Sobre-Muestreo	Sub-Muestreo	OSS	Muestreo Mixto
Musk1 [C-kNN]	**91,51**	49,89	84,37	90,64	64,4
Musk2 [mi-SVM]	**91,24**	-	91,99	90,69	91,11
Elephant [mi-SVM]	**91,76**	91,03	90,55	-	91,11
Fox [MILES]	**73,72**	64,36	68,41	72,39	67,53
Tiger [MILES]	88,62	77,25	82,37	87,35	**87,26**
Muta1 [C-kNN]	**83,39**	67,13	78,71	-	75,72
Muta2 [mi-SVM]	**72,49**	72,38	66,56	71,98	70,9
Bird WIWR [MILBoost]	92,14	89,7	**96,39**	41,77	90,09
Bird BRCR [MILBoost]	**93,14**	89,76	92,54	83,49	91,81
Web1 [mi-SVM]	**83,27**	80,27	78,49	82,17	82,97
Web2 [mi-SVM]	84,80	80,02	82,39	83,26	**85,95**

Tabla 2. Resultados obtenidos para la métrica AUCx100

Conjuntos de datos	Original	Sobre-Muestreo	Sub-Muestreo	OSS	Muestreo Mixto
Musk1 [C-kNN]	**87,86**	43,06	83,47	85,19	70,14
Musk2 [mi-SVM]	73,69	-	**75,36**	72,39	72,77
Elephant [mi-SVM]	77,84	74,91	**78,42**	-	76,88
Fox [MILES]	**67,56**	40,64	64,99	65,69	62,24
Tiger [MILES]	**80,02**	48,85	73,92	78,88	78,93
Muta1 [C-kNN]	**84,20**	79,74	79,98	-	81,64
Muta2 [mi-SVM]	**53,28**	45,72	51,52	52,97	48,31
Bird WIWR [MILBoost]	**84,50**	78,78	47,43	33,18	80,45
Bird BRCR [MILBoost]	**80,13**	71,37	59,4	52,93	79,07
Web1 [mi-SVM]	56,20	36,10	44,07	**59,78**	51,45
Web2 [mi-SVM]	**61,09**	30,14	46,5	57,74	56,52

Tabla 3. Resultados obtenidos para la métrica F1x100

Algunas observaciones pueden ser realizadas a partir de los resultados obtenidos. El desempeño de las técnicas de sobre-muestreo es siempre inferior al desempeño obtenido con el conjunto de datos original en la métrica AUC y en la mayoría de conjuntos de datos en la métrica F1. Esto sucede ya que al realizar una selección al azar de las instancias que son usadas como base para crear las instancias sintéticas, hay mayor probabilidad de seleccionar aquellas instancias negativas dentro de las bolsas positivas lo que genera mayor ambigüedad en ellas y dificulta la capacidad de los algoritmos MIL para separar el concepto de las instancias negativas. Por otro lado, aunque las técnicas de sub-muestreo disminuyen el número de instancias negativas en las bolsas negativas, se puede ver que los resultados de clasificación obtenidos con los conjuntos de datos originales son, en mayoría mejores, en ambas métricas. Esto se puede explicar debido a que la eliminación de instancias, en la forma como la realizan los métodos de sub-muestreo, conlleva a una pérdida de información importante para la definición de las fronteras de decisión.

Al aplicar el método OSS para balancear instancias, este no mostró mejoras con respecto a los resultados de la base de datos original, por el contrario, éste afectó negativamente al desempeño del clasificador. Finalmente, se puede decir que los métodos implementados para balancear exclusivamente las instancias no resultan efectivos por ser tan extremos, es decir agregar instancias hasta el número máximo de instancias por bolsa y eliminar instancias hasta el número mínimo de instancias por bolsa. Además no presentan mejoras significativas en las métricas presentadas debido a que no se contempla la cantidad de bolsas en las clases, ni la ambigüedad en las bolsas positivas, en donde se deberían duplicar las instancias que representan al concepto, y que son responsables de hacer la bolsa positiva, y eliminar las instancias negativas, que generan ambigüedad tanto en las bolsas positivas, como en la frontera de decisión con las bolsas negativas.

5 Conclusiones

En este trabajo se evaluó la aplicación de técnicas de balanceo SIL en conjuntos de datos MIL, con el fin de balancear estos conjuntos al nivel de instancias. Se utilizaron técnicas tanto de muestreo aleatorio, como muestreo informativo y se presentaron los resultados obtenidos en la experimentación. El muestreo aleatorio realizado fue bastante agresivo ya que se implementaron casos extremos de generación y eliminación de instancias (generando hasta el máximo número de instancias por bolsa y eliminando hasta el número mínimo de instancias por bolsas), esto aumentó el ruido en los conjuntos de datos y por tanto aumentó la ambigüedad en las bolsas positivas. Por otro lado el sub-muestreo se eliminó información importante para la definición de las fronteras de decisión, reduciendo el desempeño de los método MIL.

Se concluye entonces que los métodos de muestreo para conjuntos de datos de una sola instancia no pueden ser aplicados directamente a conjuntos de datos MIL ya que estos afectan negativamente el desempeño del clasificador. Es necesario tener en cuenta la naturaleza ambigua de las bolsas positivas para duplicar las instancias realmente positivas y eliminar las instancias negativas causantes de la ambigüedad.

Con base en lo anterior, el trabajo futuro en esta área de investigación, debe estar dirigido a desarrollar métodos de balanceo que tengan en cuenta las particularidades de los conjuntos de datos tipo MIL y los tres tipos de desbalanceo que se pueden presentar en los mismos.

Referencias

[1] J. Amores, "Multiple instance classification: Review, taxonomy and comparative study," *Artif. Intell.*, vol. 201, pp. 81–105, Aug. 2013.

[2] B. Babenko, "Multiple instance learning: algorithms and applications," 2008.

[3] J. Foulds and E. Frank, "A review of multi-instance learning assumptions," *Knowl. Eng. Rev.*, vol. 25, no. 01, pp. 1–25, 2010.

[4] Y. Sun, A. K. C. Wong, and M. S. Kamel, "Classification of Imbalanced Data: a Review," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 4, pp. 687–719, 2009.

[5] H. He and E. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009.

[6] G. H. Nguyen, S. L. Phung, and A. Bouzerdoum, "Learning pattern classification tasks with imbalanced data sets," in *Pattern Recognit*, InTech, 2009, pp. 193–208.

[7] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009.

[8] C. L. de Castro and A. P. Braga, "Aprendizado supervisionado com conjuntos de dados desbalanceados," *Sba Control. Automação Soc. Bras. Autom.*, vol. 22, no. 5, pp. 441–466, 2011.

[9] M. Kubat and S. Matwin, "Addressing the Curse of Imbalanced Training Sets: One-Sided Selection," in *ICML*, 1997, pp. 179–186.

[10] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.

[11] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *IJCNN*, 2008, pp. 1322–1328.

[12] P. Domingos, "MetaCost: a general method for making classifiers cost-sensitive," in *SIGKDD*, 1999, pp. 155–164.

[13] N. Thai-Nghe, Z. Gantner, and L. Schmidt-Thieme, "Cost-sensitive learning methods for imbalanced data," in *IJCNN*, 2010, pp. 1–8.

[14] H. Lee and S. Cho, "The novelty detection approach for different degrees of class imbalance," in *ICONIP - Part II*, 2006, pp. 21–30.

[15] D. M. J. Tax and R. P. W. Duin, "Support Vector Data Description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, 2004.

[16] H. Guo and H. L. Viktor, "Learning from imbalanced data sets with boosting and data generation: the DataBoost-IM approach," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 30–39, 2004.

[17] X.-Y. Liu and Z.-H. Zhou, "The Influence of Class Imbalance on Cost-Sensitive Learning: An Empirical Study," in *ICDM*, 2006, pp. 970–974.

[18] S. Wang, "Ensemble diversity for class imbalance learning," Ph.D. Thesis, The University of Birmingham, 2011.

[19] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," in *ICML*, 2007, pp. 935–942.

[20] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, 2004.

[21] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning," in *Advances in Intelligent Computing*, vol. 3644, Springer Berlin Heidelberg, 2005, pp. 878–887.

[22] I. Tomek, "Two Modifications of CNN," *IEEE Trans. Syst. Man. Cybern.*, vol. 6, no. 11, pp. 769–772, 1976.

[23] P. Hart, "The condensed nearest neighbor rule (Corresp.)," *IEEE Trans. Inf. Theory*, vol. 14, no. 3, pp. 1966–1967, 1968.

[24] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection," *ICML*, vol. 4, 1997.

[25] S. J. Yen and Y. S. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," *Expert Syst. Appl.*, vol. 36, no. 3 PART 1, pp. 5718–5727, 2009.

[26] S. García and F. Herrera, "Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy," *Evol. Comput.*, vol. 17, no. 3, pp. 275–306, 2009.

[27] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, no. 1–2, pp. 31–71, 1997.

[28] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Adv. Neural Inf. Process Syst.*, 2003, pp. 561–568.

[29] P. Viola, J. C. Platt, and C. Zhang, "Multiple Instance Boosting for Object Detection," in *Adv. Neural Inf. Process Syst.*, 2005, pp. 1417–1426.

[30] Y. Chen, J. Bi, and J. Z. Wang, "MILES: Multiple-Instance Learning via Embedded Instance Selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1931–1947, 2006.

[31] Z. Fu, A. Robles-Kelly, and J. Zhou, "MILIS: Multiple Instance Learning with Instance Selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 958–977, May 2011.

[32] X. Wang, X. Liu, N. Japkowicz, and S. Matwin, "Resampling and Cost-Sensitive Methods for Imbalanced Multi-instance Learning," *ICDM*, pp. 808–816, Dec. 2013.

[33] C. Mera, M. Orozco-alzate, and J. Branch, "Improving Representation of the Positive Class in Imbalanced Multiple-Instance Learning," *ICIAR.*, vol. 8814, pp. 266–273, 2014.

[34] R. Srinivasan, Ashwin and Muggleton, S and King, "Comparing the use of background knowledge by two Inductive Logic Programming systems," in *Inductive logic Programming: 7th International Workshop*, 1995.

[35] Z.-H. Zhou, K. Jiang, and M. Li, "Multi-Instance Learning Based Web Mining," *Appl. Intell.*, vol. 22, no. 2, pp. 135–147, 2005.