Preprocesamiento de Datos mediante el Incremento de la Tasa de Sobremuestreo para Problemas de *Big Data* Extremadamente Desbalanceados

Sara del Río, José Manuel Benítez, and Francisco Herrera

Departamento de Ciencias de la Computación e Inteligencia Artificial, CITIC-UGR (Centro de Investigación en Tecnologías de la Información y las Comunicaciones), Granada, España,
{srio,j.m.benitez,herrera}@decsai.ugr.es

Resumen El término "*big data*" ha llamado la atención de los expertos en el contexto del aprendizaje a partir de datos. Este término es usado tanto para describir el crecimiento exponencial de los datos como la disponibilidad de los mismos (estructurados y no estructurados). El diseño de modelos efectivos que puedan procesar y extraer conocimiento a partir de dichos datos representa un inmenso desafío. Por otra parte, muchas aplicaciones del mundo real presentan una distribución de datos compleja donde las clases se encuentran desbalanceadas. En este trabajo, se analiza una hipótesis con el objetivo de incrementar la precisión de las clases menos representadas en problemas de *big data* extremadamente desbalanceados. Se presenta un estudio experimental sobre el problema de *big data* extremadamente desbalanceado utilizado en la competición *ECBDL14 Big Data Competition*. Los resultados obtenidos muestran que es necesario encontrar un equilibrio entre las clases con el fin de obtener una mejor precisión.

Palabras clave: *Big Data*, *Hadoop*, *MapReduce*, Preprocesamiento, Muestreo de Datos, Datos Desbalanceados, *Random Forest*

1 Introducción

En la actualidad, uno de los mayores desafíos y retos en tecnologías de la información es el procesamiento eficiente de grandes cantidades de datos con el objeto de obtener información valiosa que ayude a la toma de decisiones en distintas áreas. Estas cantidades de datos son popularmente conocidas como *big data* [1][2]. Debido a que las técnicas y herramientars tradicionales no son capaces de hacer frente a problemas de *big data*, están apareciendo numerosas soluciones para el análisis y la gestión de los datos. Estas enormes cantidades de información también afectan a los métodos de minería de datos tradicionales, que necesitan ser adaptados para tratar tales cantidades de datos [3] [4].

Uno de los retos que dificulta la extración de conocimiento es el problema de clasificación sobre conjuntos de datos desbalanceados [5] [6]. Esta situación se produce cuando existe una desproporción notable en el número de ejemplos

pertenecientes a cada clase. Este problema ha ganado mucha importancia en los últimos años ya que se encuentra presente en muchas aplicaciones reales tales como finanzas o diagnósis médica. En estos casos, el interés de los expertos se centra en la deteción de las clases menos representadas. Los problemas de *big data* también están afectados por este desbalanceo de clases.

Para tratar de abordar de forma eficiente problemas de *big data* han aparecido numerosas soluciones, siendo una de las más relevantes el modelo de programación denominado *MapReduce* [7]. Este modelo de programación divide el conjunto de datos original en pequeños subconjuntos que son procesados en paralelo de forma independinete y a continuación combinados para obtener una solución final. No obstante, esta división de los datos puede provocar un efecto negativo en dominios desbalanceados. Entre los factores que pueden degradar el rendimiento en clasificación podemos encontrar el problema de la falta de densidad, relacionado con el tamaño del conjunto de entrenamiento [6]. Este problema se amplifica cuando la clase minoritaria tiene una representación baja, ya que provoca la aparición de los *small disjuncts* cuando el conjunto de datos original se divide y distribuye por el procedimiento *MapReduce* [8] [9].

En [10] los autores compararon varias técnicas tales como sobremuestreo, bajomuestreo o aprendizaje sensible al coste, adaptadas para abordar problemas de *big data* desbalanceados usando *MapReduce*. Una de las conclusiones de este estudio fue que la técnica de sobremuestreo era más robusta que el resto cuando se incrementaba el número de particiones sobre los datos. El bajo rendimiento del resto de técnicas se debe en gran parte al problema de la falta de densidad, agrabado por las divisiones realizadas sobre los datos originales. Además, cuando el número de particiones es elevado el número de ejemplos de la clase minoritaria es considerablemente más pequeño, agravándose aún más dicho problema.

En este trabajo, se analiza una hipótesis para abordar problemas de *big data* extremadamente desbalanceados incrementando la presencia de las clases menos representadas. Debido al problema de la falta de densidad de dichas clases y que dicho problema se ve agravado por la división de los datos que llevan a cabo los enfoques *MapReduce*, nuestra hipótesis establece que el uso de altos ratios de sobremuestreo podría mejorar el rendimiento en precisión.

Para evaluar el rendimiento de la propuesta se hace uso del problema de *big data* extremadamente desbalanceado que fue utilizado en la competición *ECBDL14 Big Data Competition* [11]. Se usan las versiones *MapReduce* de las técnicas de sobremuestreo y bajomuestreo presentadas en [10] para balancear la distribución de clases del conjunto de datos. Además, puesto que este conjunto posee un número muy elevado de características, también se utiliza la implementación *MapReduce* para selección de características basada en el empleo de esquemas de pesos [12]. Como clasificador se ha considerado la versión *MapReduce* del algoritmo *Random Forest* [13].

Este trabajo se organiza de la siguiente forma. En primer lugar, en la sección 2 se presenta una breve introducción a *big data*, conjuntos de datos desbalanceados y una breve descripción del problema utilizado en la competición *ECBDL14 Big*

Data Competition. En la sección 3 se presentan los experimentos y resultados obtenidos. Finalmente, en la sección 4 se ofrecen algunas conclusiones.

2 Preliminares

Esta sección proporciona una introducción a *big data* y al modelo de programación *MapReduce* (Sección 2.1). A continuación, la Sección 2.2 ofrece una descripción de los problemas de clasificación con conjuntos de datos descabalanceados. Finalmente, la Sección 2.3 describe el problema utilizado en la competición *ECBDL14 Big Data Competition*.

2.1 *Big data* y el Modelo de Programación *MapReduce*

Big data es un término que hace referencia a cantidades de datos que no pueden ser procesadas por técnicas y herramientras tradicionales [4]. Inicialmente, el analista Douglas Laney's de la consultora *Gartner* definió este concepto como un modelo de tres "UVES" ("Vs") denominadas Volumen, Velocidad y Variedad. El término "Volumen" hace refrencia a las enormes cantidades de información que necesitan ser procesadas y analizadas para obtener conocimiento útil y valioso. El término "Velocidad" se refiere a que los datos deben ser procesados manteniendo unos tiempos de respuesta aceptables. Finalmente, el término "Variedad" hace referencia a la diversidad y mutabilidad de los datos. Más recientemente han ido apareciendo "Vs" adicionales para completar la definición de *big data*. Algunas de ellas son Variabilidad, Veracidad, Volatilidad, Validez o Valor [4].

Una de las soluciones más populares para abordar problemas de *big data* es *MapReduce* [7]. Se trata de un modelo de programación presentado por Google en 2004 para el procesamiento de grandes cantidades de datos en clústers de nodos. *MapReduce* consta de dos fases, denominadas "*Map*" y "*Reduce*". En términos generales, en la fase *Map* los datos se dividen en conjuntos más pequeños que son distribuidos y procesados en paralelo. A continuación, en la fase *Reduce* se combinan los resultados obtenidos en la fase anterior para generar la salida final. Una de las implementaciones más populares de *MapReduce* es *Hadoop* [14]. Se trata de *framework* de libre distribución escrito en Java que permite la escritura de aplicaciones distribuidas. *Hadoop* cuenta con un sistema de ficheros distribuido denominado *Hadoop Distributed File System* (HDFS).

2.2 Clasificación con Conjuntos de Datos Desbalanceados

Muchos problemas del mundo real presentan una distribución de clases donde una o varias clases están representadas por un gran número de ejemplos con respecto al insignificante número de ejemplos del resto de las clases. Esta circunstancia es conocida como el problema de clasificación con conjuntos de datos desbalanceados y se encuentra presente en numerosas aplicaciones reales tales como dignósis médica, bioinformática o finanzas. En estos problemas, el interés de los expertos se centra en la identificación de las clases menos representadas, ya que suelen ser las más importantes desde el punto de vista del aprendizaje.

La relación de desequilibrio o *imbalance ratio* (IR), que se define como la relación entre el número de instancias de la clase mayoritaria y la clase minoritaria, permite mostrar el nivel de dificultad asociado a un conjunto de datos

específico. Por otra parte, existen factores adicionales que influyen de forma negativa en la clasificación con conjuntos de datos desbalanceados. Estos factores incluyen la existencia de una muestra de pequeño tamaño, la presencia de ruido, el solapamiento de clases o las diferencias en la distribucición de los datos entre los conjuntos de entrenamiento y prueba [6] [9] [15].

Numerosas técnicas se han propuesto para tratar con el problema de clasificación con conjuntos de datos desbalanceados [6]. Estas técnicas suelen dividirse en dos grupos: a nivel de datos y a nivel de algoritmo. Las técnicas a nivel de datos modifican el conjunto de entrenamiento original para obtener una distribución de clases equitativa. Estas técnicas suelen dividirse a su vez en dos grupos: métodos de sobremustreo, que se basan en la replicación o generación de ejemplos de la clase minoritaria, y métodos de bajomuestreo, que se centran en la eliminación de ejemplos de la clase mayoritaria. Por otra parte, las técnicas a nivel de algortimo realizan modificaciones en el funcionamiento de los algoritmos tratando de beneficiar a la clase minoritaria. Las técnicas de aprendizaje sensible al coste combinan los enfoques anteriores [16].

2.3 Conjunto de datos *ECBDL14 Big Data Competition*

Para evaluar la hipótesis se ha seleccionado el conjunto de datos que fue utilizado en la competición *ECBDL14 Big Data Competition* [11], que representa un problema de predicción de mapas de contactos dentro del ambito de la bioinformática. Este problema se ha convertido en un desafío en el campo de la predicción de la estructura de las proteinas debido a la baja densidad de los contactos (ejemplos de la clase minoritaria) y a la gran cantidad de datos que pueden extraerse a partir de tan sólo unos miles de proteinas (ejemplos de la clase mayoritaria) [17]. Este conjunto de datos cuenta con un conjunto de entrenamiento compuesto de aproximadamente 32 millones de ejemplos y un conjunto de prueba de casi tres millones de ejemplos. Además, este problema cuenta con 631 características y 2 clases, donde más del 98% son ejemplos de la clase mayoritaria y menos del 2% son contactos.

3 Análsis de la Eficacia en Preprocesamiento de Problemas de *Big Data* Extremadamente Desbalanceados

El objetivo de esta sección es analizar la eficacia del preprocesamiento cuando se trabaja con problemas de *big data* extremadamente desbalaneados. Para ello, se ha seguido el siguiente esquema de trabajo:

1. **Paso 1**: Análisis de técnicas de muestreo clásicas, tales como sobremuestreo o bajomuestro, para obtener una distribución equitativa de clases.
2. **Paso 2**: Análisis de la técnica de sobremuestreo con diferentes ratios de sobremuestreo para incrementar la tasa de verdaderos positivos. El problema de la falta de datos de la clase minoritaria es inherente a la mayoría de los problemas desbalanceados y dicho problema se ve agravado por la división de los datos que se lleva a cabo por un enfoque *MapReduce*. Por este motivo,

podría ser interesante incrementar el ratio de sobremuestreo con el fin de incrementar la tasa de verdaderos positivos.

Esta sección se organiza de la siguiente forma. En primer lugar, en la Sección 3.1 se presentan los algortimos, parámetros utilizados y las métricas empleadas para evaluar el rendimiento de la propuesta. En las Secciones 3.2 y 3.3 se presentan y analizan los resultados obtenidos para cada uno de los pasos de preprocesamiento, incluyéndose el caso de selección de características en la Sección 3.4.

3.1 Marco Experimental

Como ya se ha mencionado, se utilizan las versiones *MapReduce* de las técnicas de sobremuestreo (ROS-BigData) y bajomuestreo (RUS-BigData) presentadas en [10] para abordar el problema del desbalanceo de clases. Además, se utiliza la implementación *MapReduce* para selección de características basada en el empleo de esquemas de pesos (DEFW-BigData) [12] para seleccionar las características más relevantes. Como clasificador se ha considerado la implementación *MapReduce* del algoritmo *Random Forest* (RF-BigData), disponible en *Mahout* [13].

En la Tabla 1 se muestra la configuración de parámetros utilizada en los experimentos. Para el algoritmo ROS-BigData el parámetro *tasaSobremuestreo* representa el ratio de sobremuestreo utilizado para incrementar la proporción de ejemplos de la clase minoritaria. El algoritmo RF-BigData cuenta con los parámetros *maxProfundidad*, *numCaracteristicas* y *numArboles*, donde *maxProfundidad* indica la profundidad de cada uno de los árboles a generar, *numCaracteristicas* es el número de atributos seleccionados para construir los árboles y *numArboles* se corresponde con el número de árboles que componen el *ensemble* de árboles. Para todos los algoritmos, el parámetro *numMaps* representa el número de particiones o subconjuntos que se van a generar a partir del conjunto de datos original. Para evaluar la eficiencia en clasificación de la metodología

Tabla 1: Parámetros para los algoritmos utilizados en la experimentación

Algoritmo	Parámetros
RUS-BigData	numMaps = 1024
ROS-BigData	tasaSobremuestreo = 100, 105, 115, 130, 140, 150, 160, 170, numMaps = 1024
RF-BigData	maxProfundidad = ilimitada, numCaracteristicas = 10, 25, numMaps = 64/192, numArboles = 192

propuesta se utilizan tres medidas: **tasa de verdaderos positivos**, que se corresponde con el procentaje de ejemplos de la clase minoritaria correctamente clasificados $VP_{tasa} = \frac{VP}{VP+FN}$; la **tasa de verdaderos negativos**, que representa el porcentaje de ejemplos de la clase mayoritara correctamente clasificados $VN_{tasa} = \frac{VN}{FP+VN}$; y el producto de ambas ($VP_{tasa} \cdot VN_{tasa}$).

3.2 Sobremuestreo y Bajomuestreo

En primer lugar, se analizan los resultados obtenidos por el algoritmo RF-BigData sobre el conjunto de entrenamiento original (sin preprocesamiento).

Después, se analizan los resultados obtenidos por el algoritmo RF-BigData sobre el conjunto de entrenamiento ya balanceado, generado con las técnicas clásicas de muestreo para *big data*: ROS-BigData y RUS-BigData. En la Tabla 2 se muestran los resultados obtenidos para el conjunto de prueba utilizando 64 y 192 *maps*. El valor destacado en negrita se corresponde con el mejor resultado.

Tabla 2: Resultados obtenidos usando 64 y 192 *maps* y 10 características internas para RF-BigData

Algoritmo	*Maps*	VP_{tasa}	VN_{tasa}	$VP_{tasa} \cdot VN_{tasa}$
RF-BigData	64	0,000000	1,000000	0,000000
	192	0,000000	1,000000	0,000000
RUS-BigData + RF-BigData	64	0,641076	0,753291	0,482917
	192	0,636717	0,748135	0,476350
ROS-BigData (100%) + RF-BigData	64	0,598474	0,815745	0,488202
	192	0,617061	0,791892	**0,488646**

De acuerdo con los resultados se pueden extraer las siguientes conclusiones:

- El uso del algoritmo RF-BigData sobre el conjunto de entrenamiento original proporciona resultados fuertemente sesgados a favor de la clase mayoritaria. Por ello, es absolutamente necesaria la aplicación de técnicas de muestreo.
- RUS-BigData proporciona peores resultados que ROS-BigData en problemas altamente desbalanceados. Esto fue estudiado en detalle en [10], donde se observó que la técnica de bajomuestreo sufre el problema de la muestra de pequeño tamaño, asociado a la división de los datos que lleva a cabo un enfoque *MapReduce*.
- La combinación de ROS-BigData con RF-BigData proporciona los mejores resultados. No obstante, aunque este método trabaja mejor, se pueden observar unos valores muy bajos para VP_{tasa} con respecto a los valores para VN_{tasa}. Esto podría ser debido a que, aunque ROS-BigData proporciona un gran número de ejemplos de la clase minoriatria, podría existir una presencia desbalanceada de las instancias en las particiones realizadas por el enfoque MapReduce. Por este motivo pensamos que un incremento en la tasa de sobremuestreo podría dar lugar a un incremento de los valores para la VP_{tasa} y, por lo tanto, un incremento en el rendimiento general. Por otra parte, se puede observar que con 192 *maps* se obtienen ligeramente mejores resultados con respecto a 64 *maps*. Este comportamiento puede ser debido a la distribución de los datos del conjunto de entrenamiento en los *maps*.

3.3 Sobremuestreo con Altos Ratios para Mejorar la Tasa de Verdaderos Positivos

Con el fin de sesgar al clasificador RF-BigData hacia la clase minoritaria, consideramos incrementar la densidad de dicha clase. Por ello, incrementamos poco a poco el ratio de sobremuestreo desde 105% hasta 150%. La Figura 1 muestra

el procedimiento llevado a cabo. La Tabla 3 muestra los resultados obtenidos para el conjunto de prueba utilizando 64 y 192 *maps* y tasas de sobremustreo que van desde el 105% al 150%.

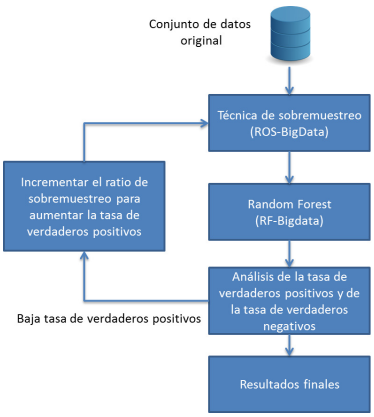

Fig. 1: Diagrama de flujo que ilustra el proceso para aumentar la tasa de verdaderos positivos

Tabla 3: Resultados obtenidos para diferentes tasas de sobremustreo usando 64 y 192 *maps* y 10 características internas para RF-BigData

	64 *maps*			192 *maps*		
$TasaSobremuestreo$	VP_{tasa}	VN_{tasa}	$VP_{tasa} \cdot VN_{tasa}$	VP_{tasa}	VN_{tasa}	$VP_{tasa} \cdot VN_{tasa}$
105%	0,619446	0,800734	0,496012	0,642762	0,774510	0,497826
115%	0,653289	0,772620	0,504744	0,681991	0,736557	0,502326
130%	0,704546	0,725117	**0,510878**	0,733803	0,685623	**0,503113**
140%	0,704710	0,720721	0,507900	0,734482	0,684857	0,503015
150%	0,722310	0,706574	0,510365	0,765323	0,649534	0,497103

A partir de los resultados obtenidos podemos observar que conforme se incrementa la tasa de sobremuestreo, los valores para la VP_{tasa} también se ven incrementados independientemente del número de particiones o *maps* utilizados. También podemos observar que cuando aumentan los valores para la VP_{tasa}, disminuyen los valores para la VN_{tasa}. Por ello, es necesario encontrar un equilibrio entre dichos valores con el objetivo de obtener la máxima precisión en clasificación ($VP_{tasa} \cdot VN_{tasa}$). En este caso, hemos encontrado un equilibrio en el rendimiento de ambas clases cuando se utiliza un ratio de sobremuestreo del 130%, tanto para 64 como para 192 *maps*.

En este punto queremos comparar el mejor resultado obtenido hasta el momento con el mejor resultado obtenido por el segundo y tercer puesto en la competición *ECBDL14 Big Data Competition* (ver Tabla 4). Se puede observar que el mejor resultado obtenido hasta este punto no se encuentra demasiado lejos de los resultados obtenidos por los participantes que obtuvieron el segundo y tercer puesto. Además, se puede ver que nuestro mejor resultado, obtenido usando ROS-BigData con una tasa de sobremuestreo del 130% y 64 *maps*, se encuentra por encima de los resultados obtenidos por el tercer puesto.

Tabla 4: Comparación con el segundo y tercer puesto en la competición *ECBDL'14 Big Data Competition*

Algoritmo/Equipo	VP_{tasa}	VN_{tasa}	$VP_{tasa} \cdot VN_{tasa}$
ICOS (2°)	0,703210	0,730155	**0,513452**
ROS-BigData (130%) + RF-BigData	0,704546	0,725117	0,510878
UNSW (3°)	0,699159	0,727631	0,508730

3.4 Sobremuestreo con Altos Ratios y Selección de Características

Puesto que el conjunto de datos utilizado en la competición *ECBDL14 Big Data Competition* contiene un número elevado de características, hemos decidido utilizar un nuevo componente de preprocesamiento para mejorar el rendimiento en clasificación mediante la obtención de las características más relevantes. Para ello utilizamos el algoritmo DEFW-BigData, que calcula la importancia de cada una de las características en términos de pesos. DEFW-BigData genera un vector de pesos a partir del cual se seleccionan las características cuyo peso supere un cierto umbral. En este trabajo se ha seleccionado dicho umbral a partir de resultados obtenidos en experimentos preliminares que no se muestran en este trabajo. Al final de este proceso se ha obtenido un conjunto de 90 de las 631 características.

Una vez seleccionadas las características, hemos repetido la experimentación utilizando el algoritmo ROS-BigData con diferentes tasas de sobremuestreo desde 100% hasta 150%. También hemos incrementado el número de características internas utilizadas por RF-BigData de 10 a 25. La Tabla 5 muestra los resultados obtenidos para el conjunto de prueba utilizando 90 características y 64 *maps*. Utilizamos este número de particiones ya que con dicho número se obtuvieron los mejores resultados en la sección anterior (ver Tabla 3). A partir de los resul-

Tabla 5: Resultados obtenidos con selección de características y diferentes tasas de sobremustreo usando 64 *maps* y 25 características internas para RF-BigData

	64 *maps*		
$TasaSobremuestreo$	VP_{tasa}	VN_{tasa}	$VP_{tasa} \cdot VN_{tasa}$
100%	0,621728	0,822059	0,511097
130%	0,671279	0,783911	0,526223
140%	0,695109	0,763951	0,531029
150%	0,705882	0,753625	**0,531971**

tados obtenidos podemos observar que el uso de un conjunto de características más reducido nos ha permitido obtener una mayor precisión en comparación con los resultados de la sección anterior. También podemos observar que el algoritmo DEFW-BigData ha permitido incrementar los valores para la VP_{tasa} pero también los valores para la VN_{tasa}, produciéndose un desbalanceo en la precisión obtenida en ambas clases. Hay que tener en cuenta que en la sección anterior obtuvimos un equilibrio en el rendimiento de ambas clases con una tasa de sobremuestreo de 130% (ver Tabla 3). El uso de DEFW-BigData ha permitido obtener un mejor rendimiento en precisión pero también ha provocado la

aparición de altas diferencias entre la precisión obtenida para la clase mayoritaria y minoritaria. Por este motivo, considerando las conclusiones obtenidas en el apartado anterior, aumentamos aún más la tasa de sobremuestreo con el fin de encontrar un equilibrio entre ambas clases. La Tabla 6 muestra los resultados obtenidos para el conjunto de prueba utilizando 90 características, altas tasas de sobremuestreo y 64 *maps*. A partir de los resultados podemos concluir que es

Tabla 6: Resultados obtenidos con selección de características y tasas de sobremustreo elevadas usando 64 *maps* y 25 características internas para RF-BigData

	64 *maps*		
$TasaSobremuestreo$	VP_{tasa}	VN_{tasa}	$VP_{tasa} \cdot VN_{tasa}$
160%	0,718692	0,741976	0,533252
170%	0,730432	0,730183	**0,533349**
180%	0,737381	0,722583	0,532819

necesario el uso de altos ratios de sobremuestreo para encontrar un equilibrio en el rendimiento de ambas clases. En este caso, hemos encontrado dicho equilibrio con un ratio de sobremuestreo del 170%.

Finalmente, la Tabla 7 muestra los tres mejores resultados de la competición *ECBDL'14 Big Data Competition*.

Tabla 7: Resultados de los tres primeros puestos de la competición *ECBDL'14 Big Data Competition*

	64 *maps*		
$Algoritmo/Equipo$	VP_{tasa}	VN_{tasa}	$VP_{tasa} \cdot VN_{tasa}$
Efdamis (1°)	0,730432	0,730183	**0,533349**
ICOS (2°)	0,703210	0,730155	0,513452
UNSW (3°)	0,699159	0,727631	0,508730

4 Comentarios Finales

En problemas de clasificación con datos desbalaneados la falta de densidad de la clase minoritaria provoca un impacto negativo en rendimiento. En *big data*, dicho impacto es aún mayor cuando un enfoque *MapReduce* particiona el conjunto de datos original en conjuntos más pequeños. Por este motivo, analizamos una hipótesis que establece que un incremento de la densidad de la clase minoritaria mediante el uso de altas tasas de sobremuestreo podría mejorar el rendimiento.

El estudio experimental llevado a cabo sobre el conjunto de datos utilizado en la competición *ECBDL'14 Big Data Competition* soporta dicha hipótesis, mostrándose una mejora de rendimiento. Además, señala la necesidad de establecer la tasa de sobremuestreo a un valor que permita obtener un equilibrio entre los valores para VP_{tasa} y VN_{tasa} con el fin de obtener los mejores resultados en términos de precisión.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad bajo el proyecto TIN2014-57251-P y por el Plan Andaluz de Investigación bajo los proyectos P11-TIC-7765 y P10-TIC-6858.

Referencias

1. Zikopoulos, P., Eaton, C., DeRoos, D., Deutsch, T., Lapis, G.: Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. McGraw-Hill. (2011)
2. Madden, S.: From Databases to Big Data. IEEE Internet Computing, 16, 3, 4–6 (2012)
3. Wu, X., Zhu, X., Wu, G.Q., Ding, W.: Data Mining with Big Data. IEEE Transactions on Knowledge and Data Engineering. 26, 1, 97–107 (2014)
4. Fernández, A., Río, S., López, V., Bawakid, A., del Jesus, M.J., Benítez, J. M., Herrera, F.: Big Data with Cloud Computing: An Insight on the Computing Environment, MapReduce and Programming Frameworks. WIREs Data Mining and Knowledge Discovery. 4, 5, 380–409 (2014)
5. He, H., García, E. A.: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering. 21, 9, 1263–1284 (2009)
6. López, V., Fernández, A., García, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information Sciences. 250, 113–141 (2013)
7. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Communications of the ACM. 51, 1, 107–113 (2008)
8. Weiss, G.M., Provost, F.J.: Learning when training data are costly: the effect of class distribution on tree induction. Journal of Artificial Intelligence Research. 19, 315–354 (2003)
9. Weiss, G.M.: Mining with rarity: a unifying framework. SIGKDD Explorations. 6, 1, 7–19 (2004)
10. Río, S., López, V., Benítez, J.M., Herrera, F.: On the use of MapReduce for imbalanced big data using Random Forest. Information Sciences. 285, 0, 112–137 (2014)
11. ECBDL'14 Big Data Competition. [Online; consultado en Septiembre 2015]. http://cruncher.ncl.ac.uk/bdcomp/. (2014)
12. Triguero, I., Río, S., López, V., Bacardit, J., Benítez, J.M., Herrera, F.: ROSEFW-RF: The winner algorithm for the ECBDL'14 Big Data Competition: An extremely imbalanced big data bioinformatics problem. Knowledge-Based Systems. 87, 69–79 (2015)
13. Apache Mahout Project. [Online; consultado en Septiembre 2015]. http://mahout.apache.org/. (2015)
14. White, T.: Hadoop, The Definitive Guide. O'Reilly Media, Inc. (2012)
15. García, V., Mollineda, R.A., Sánchez, J.S.: On the k-NN performance in a challenging scenario of imbalance and overlapping. Pattern Analysis and Applications. 11, 3–4, 269–280 (2008)
16. López, V., Fernández, A., Moreno-Torres, J.G., Herrera, F.: Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics. Expert Systems with Applications. 39, 7, 6585–6608 (2012)
17. Bacardit, J., Widera, P., Marquez-Chamorro, A., Divina, F., Aguilar-Ruiz, J.S. Krasnogor, N.: Contact map prediction using a large-scale ensemble of rule sets and the fusion of multiple predicted structural features. Bioinformatics. 28, 19, 2441–2448 (2012)