Analysis of Data Preprocessing Increasing the Oversampling Ratio for Extremely Imbalanced Big Data Classification

Sara del Río
Department of Computer Science
and Artificial Intelligence, CITIC-UGR
(Research Center on Information
and Communications Technology).
University of Granada, 18071
Granada, Spain
Email: srio@decsai.ugr.es

José M. Benítez
Department of Computer Science
and Artificial Intelligence, CITIC-UGR
(Research Center on Information
and Communications Technology).
University of Granada, 18071
Granada, Spain
Email: J.M.Benitez@decsai.ugr.es

Francisco Herrera
Department of Computer Science
and Artificial Intelligence, CITIC-UGR
(Research Center on Information
and Communications Technology).
University of Granada, 18071
Granada, Spain
Email: herrera@decsai.ugr.es

***Abstract*—The "big data" term has caught the attention of experts in the context of learning from data. This term is used to describe the exponential growth and availability of data (structured and unstructured). The design of effective models that can process and extract useful knowledge from these data represents a immense challenge. Focusing on classification problems, many real-world applications present a class distribution where one or more classes are represented by a large number of examples with respect to the negligible number of examples of other classes, which are precisely those of primary interest. This circumstance is known as the problem of classification with imbalanced datasets. In this work, we analyze a hypothesis in order to increment the accuracy of the underrepresented class when dealing with extremely imbalanced big data problems under the MapReduce framework. The performance of our solution has been analyzed in an experimental study that is carried out over the extremely imbalanced big data problem that was used in the ECBDL'14 Big Data Competition. The results obtained show that is necessary to find a balance between the classes in order to obtain the highest precision.**

I. Introduction

In recent years, one of the biggest challenges in information technology is the efficient processing of huge amounts of data that grow day after day, with the aim of obtaining valuable information that can be used for decision-making in different areas. These amounts of data are referred to as "big data" [1] [2]. Since the traditional tools and techniques are not able to address such amounts of data, new solutions for data management and data analysis are emerging. These huge amounts of information also affect to data mining and machine learning algorithms as they need to be adapted in order to cope with this challenge [3] [4].

One of the challenges that hinders the extraction of useful knowledge is the problem of classification with imbalanced datasets [5] [6]. This problem occurs when the number of examples of one or more classes is essentially smaller than the number of instances belonging to the other classes. The relevance of this problem lies in its existence in many real-world applications such as finance or medical diagnosis, among many others. In these cases, the interest of experts focuses on the detection of the less representative classes. Big data is also impacted by this class imbalance.

In order to effectively address big data problems, numerous solutions have emerged being MapReduce one of the most popular [7]. It is a programming model which divides the original data into smaller subsets that are processed independently in parallel, and whose partial solutions are then combined in order to obtain a final solution. However, this division of the data may have a negative effect on imbalanced domains. Among the difficulties that may degrade the performance in classification with imbalanced datasets, we can find the problem called "lack of density" or "lack of data" related to the training-set size [6]. It is amplified when the minority class has a low representation, due to it leads to the appearance of small disjuncts with the MapReduce data fragmentation [8] [9].

In [10] the authors compared several techniques such as random oversampling, artificial random oversampling, random undersampling and cost-sensitive learning, which were adapted to address imbalanced big data using MapReduce. One of the findings of this study was that random oversampling was more robust than the other techniques when the number of partitions is increased. The poor performance of random undersampling and cost-sensitive learning methods is mainly due to the small sample size problem, which is aggravated by the splitting of the original data. Moreover, when the number of partitions is elevated, the number of minority class examples is considerably smaller and, therefore the lack of density is amplified.

In this work, we analyze a hypothesis to deal with extremely imbalanced big data problems increasing the presence of the underrepresented class. Due to the problem of the lack of density of the underrepresented class, aggravated by the splitting of data that is performed in the MapReduce approaches, our hypothesis states that the use of high oversampling ratios could improve the performance results.

In order to evaluate the performance of our solution, we used the extremely imbalanced big data problem used in the ECBDL'14 Big Data Competition [11] with nearly 32 million examples and 631 features. We use the MapReduce versions of the random oversampling and the random undersampling techniques presented in [10] in order to balance the highly imbalanced class distribution of the dataset. Furthermore, due to the large number of features that this dataset has, we also apply the MapReduce approach for evolutionary feature weighting introduced in [12] with the aim of detect the most significant features. We use as base classifier a MapReduce implementation of the Random Forest algorithm [13].

The rest of this work is organized as follows. In Section II some background information about big data, imbalanced datasets and a brief description of the Bioinformatics problem utilized in the ECBDL'14 Big Data Competition are provided. Section III presents the experimental study conducted, detailing information about the experiments configuration, the results obtained and an analysis of them. Finally, Section IV shows the conclusions achieved in this work.

II. Preliminaries

In this section we present the context in which this paper is focused. First, in section II-A, we provide an introduction about big data and the MapReduce programming model. Then, in section II-B, we describe the problem of classification with imbalanced data. Finally, in section II-C we provide a description about the Bioinformatics problem used in the ECBDL'14 Big Data Competition.

A. Big Data and the MapReduce Programming Model

"Big data" is a term used to describe huge amounts of data so large and complex that cannot be processed by traditional tools and techniques in an easy way [4]. Initially, Douglas Laney's Gartner analyst defined this concept as a three Vs model (Volume, Velocity and Variety), where "Volume" refers to the vast amounts of data that needs to be processed and analyzed in order to obtain valuable information, "Velocity" states that the data must be processed in an acceptable response time, and finally, "Variety" means that the data can be presented in different formats. Later, additional Vs have been introduced to expand the description of the "big data" term, and some of these characteristics are Variability, Veracity, Volatility, Validity or Value [4].

One of the best known solutions to address big data problems is MapReduce [7], a parallel programming model presented by Google in 2004 that allows the processing of huge amounts of data on clusters of nodes. The MapReduce programming model consists of two phases, called "Map" and "Reduce". In general terms, the Map phase split the data into smaller subsets that are distributed through parallel processing nodes and processed in parallel. Then, in the Reduce phase, the results generated in the previous phase are collected and combined in some way to produce the final output.

More specifically, MapReduce is based on a basic structure of pairs (key, value). In the Map phase each node applies in parallel a Map function "Map()" to each pair of data from its partition and produces a list of pairs that are stored in a temporary storage. Between the Map and Reduce phases there is a phase called "Shuffle" which is responsible for grouping the pairs produced by the Map function with the same key. Finally, in the Reduce phase each node applies in parallel a Reduce function "Reduce()" to each group generated in the previous phase and produce the corresponding pair as the final output. Figure 1 shows a typical MapReduce execution with its "Map()" (denoted as *M*) and "Reduce()" (denoted as *R*) functions. The terms *k* and *v* refer to the key and value pair respectively.

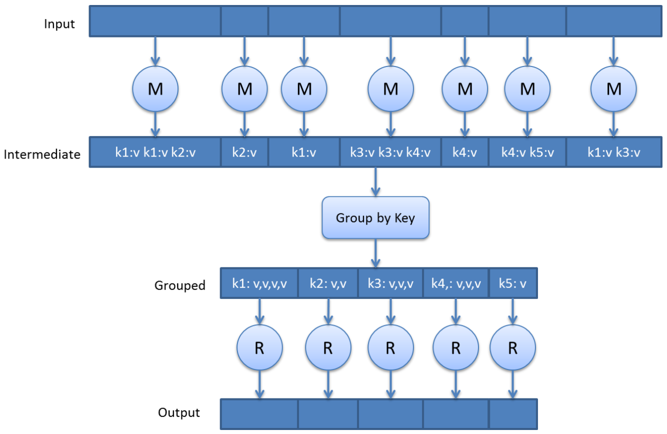

Fig. 1. The MapReduce execution scheme

Since the MapReduce technology is owned by Google and is not available for public use, an implementation of this framework called Hadoop [14] was adopted by the Apache Software Foundation. Hadoop is an open source Java project for writing distributed applications that process large amounts of data on clusters of nodes. Furthermore, Hadoop also implements a distributed file system, called Hadoop Distributed File System (HDFS), which is similar to Google File System.

B. Classification with Imbalanced Datasets

Many real-world problems usually have a distribution of classes where one or more classes are represented by a large number of examples in contrast to the negligible number of examples of other classes. This is known as the problem of classification with imbalanced data and it is present in different domains such as finances, bioinformatics or medical applications. In these problems, the main concern is the correct identification of the underrepresented classes since they are the focus of interest.

The traditional classification algorithms are often unable to address imbalanced datasets as they are built under the assumption of obtaining a greater generalization ability. For this reason, these algorithms try to get general rules that cover most of the examples, benefiting the most represented classes and trying as noise the underrepresented classes [5] [6].

The imbalance ratio (IR), defined as the ratio of the number of examples in the majority class in relation to the number of examples in the minority class, allows to show the difficulty level of a specific dataset from the imbalanced point of view. On the other hand, there are some intrinsic characteristics of the data that hamper even more the learning process with imbalanced datasets. These features include the existence of small disjuncts in the data, the small sample size, the overlapping between the classes, the presence of noise, the borderline examples and the dataset shift [6] [9] [15].

Numerous approaches have been proposed to address the problem of classification with imbalanced datasets [6]. These techniques are usually classified into two groups: data-level approaches and algorithm-level approaches. The data-level approaches try to modify the original training set in order to obtain an almost balanced class distribution that can be later used by traditional learning algorithms. These approaches are often subdivided into two groups: oversampling methods and undersampling methods. The oversampling methods are based on adding examples of the minority class to balance the class distribution while undersampling methods try to adjust the class distribution by eliminating majority class examples. On the other hand, the algorithm-level approaches perform modifications to the algorithms in order to improve the classification of the instances of the underrepresented classes. Cost-sensitive learning approaches combine the ideas from both the data-level and algorithm-level approaches considering larger misclassification costs for the examples that belong to the underrepresented class and minimizing the global cost [16].

C. ECBDL'14 Big Data Competition Bioinformatic Dataset

To analyze the quality of the hypothesis we selected the dataset that was used in the ECBDL'14 Big Data Competition [11], which represents the problem of contact map prediction in the area of bioinformatics. This problem has become one of the most challenging goals in the field of protein structure prediction due to the low density of the contacts (examples of the underrepresented class) and the large amount of data extracted from only a few thousand proteins (examples of the majority class) [17]. This dataset consists of a training set composed of approximately 32 million examples and a test set of nearly three million examples. Furthermore, this problem has 631 features and 2 classes where more than 98% of the examples correspond with the majority class and less than 2% are contacts.

III. Analysis of the Effectiveness in Preprocessing Extremely Imbalanced Big Data

At this point, our goal is to analyze the effectiveness of preprocessing when working with extremely imbalanced big data problems. To do this, we will follow the work scheme described below:

1) Step 1: Analysis of the classical sampling techniques such as random oversampling and random undersampling for balancing the classes distribution.
2) Step 2: Analysis of random oversampling with different oversampling percentages in order to increment the true positive rate. The problem of the lack of data of the minority class, also known as the small sample size problem, is inherent to some imbalanced distributions and is aggravated by the division of the data that the MapReduce process carries out. For this, could be interesting to increase the oversampling ratio in order to enhance the true positive rate.

This section is organized as follows. First, in Section III-A we describe the algorithms and their configuration parameters, the metrics used to evaluate the performance of the solution and the infrastructure utilized. Next, in Section III-B and Section III-C we show and discuss the performance results we obtained for each preprocessing step, including the case of feature selection in Section III-D.

A. Experimental Framework

As we have mentioned, we use the MapReduce versions for the random oversampling algorithm (ROS-BigData) and the random undersampling algorithm (RUS-BigData) presented in [10] to address the class imbalanced problem. Additionally, we use the MapReduce approach for evolutionary feature weighting (DEFW-BigData)[12], which was used in the ECBDL'14 Big Data Competition, for detecting of the most important features. As classifier we use the MapReduce version of the Random Forest algorithm available in the Mahout library [13] (RF-BigData).

The configuration parameters used for the preprocessing experiments are shown in Table I. For the ROS-BigData algorithm the *oversamplingPercentage* parameter represents the oversampling rate used to increase the proportion of positive examples in the resulting preprocessed dataset. The RF-BigData algorithm is run using the *maxDepth*, *numFeatures* and *numTrees* parameters, where *maxDepth* corresponds with the depth of the trees generated, *numFeatures* indicates the number of selected attributes to build the trees and *numTrees* corresponds with the number of trees that compose the forest. For all algorithms, the *numMaps* parameter represents the number of subsets of the original data that are created.

TABLE I. PARAMETER SPECIFICATION FOR THE ALGORITHMS TESTED IN THE EXPERIMENTATION

Algorithm	Parameters
RUS-BigData	numMaps = 1024
ROS-BigData	oversamplingPercentage = 100, 105, 115, 130, 140, 150, 160, 170, numMaps = 1024
RF-BigData	maxDepth = unlimited, numFeatures = 10, 25, numMaps = 64/192, numTrees = 192

The effectiveness in classification for the proposed methodology will be evaluated using three measures: the **true positive rate**, which is the percentage of positive examples correctly classified $TP_{rate} = \frac{TP}{TP+FN}$; the **true negative rate**, which is the percentage of negative examples correctly classified $TN_{rate} = \frac{TN}{FP+TN}$; and the product of both $TP_{rate} \cdot TN_{rate}$.

Regarding the infrastructure used, all the experiments have been executed on the research group's cluster which is composed of 20 nodes connected through a 40Gb/s Infiniband network. Each node has two Intel Xeon E5-2620 microprocessors (each one with 6 cores, 15MB cache at 2 GHz) and 64GB of main memory working under Linux CentOS 6.5. The head node of the cluster has two Intel Xeon E5-2620 microprocessors (at 2.00 GHz, 15MB cache) and 96GB of main memory. The cluster is configured with Hadoop 2.0.0 (Cloudera CDH4.7.1).

B. Random Oversampling and Random Undersampling

In first place, we analyze the results obtained by the RF-BigData algorithm over the original training data (without preprocessing). In a second step, we also analyze the results obtained by the RF-BigData algorithm over the balanced training data, generated with the classical data sampling techniques for big data: ROS-BigData and RUS-BigData.

Table II shows the results in test achieved using 64 and 192 maps. The value highlighted in bold corresponds to the best result.

TABLE II. RESULTS OBTAINED USING 64 AND 192 MAPS AND 10 INTERNAL FEATURES FOR RF-BIGDATA

$Algorithm$	$Maps$	TP_{rate}	TN_{rate}	$TP_{rate} \cdot TN_{rate}$
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**

According to the results, we extract the following conclusions:

- The use of the RF-BigData algorithm over the original training data provides totally biased results to the majority class. Therefore, the application of data sampling techniques is absolutely necessary.
- The results achieved by RUS-BigData show that its application in highly imbalanced problems provides worse performance than ROS-BigData. This fact was discussed in detail in [10], where it was observed that undersampling suffers from the small sample size of the underrepresented class, associated to the splitting of data performed in the MapReduce model.
- The ROS-BigData algorithm combined with RF-BigData provides the best performance results. However, although this method works best, we can observe a very low TP_{rate} compared to the TN_{rate}. This could be due to, although the ROS-BigData algorithm provides a large number of examples of the minority class, there could be an unbalanced presence of the instances in the data splits. For this reason, we think that an increase in the oversampling ratio could lead to an increase in the TP_{rate} values and therefore, an increment in overall performance.

C. Random Oversampling with Higher Oversampling Ratios to Enhance the True Positive Rate

In order to bias the RF-BigData classifier towards the minority class, we consider to increase the density of this class. We increment the oversampling ratio in small steps from 105% to 150%. Figure 2 shows the procedure that we have carried out.

Table III shows the results in test obtained over the extremely imbalanced big data problem considered in this study using 64 and 192 maps and oversampling rates from 105% to 150%. The values highlighted in bold correspond to the bests results.

We can extract the following conclusions from the results obtained:

- When the oversampling rate is increased the TP_{rate} values also increase independently of the number of maps utilized.
- In general, the $TP_{rate} \cdot TPN_{rate}$ results drop slightly as we increase the number of splits in the data (from 64 to 192).

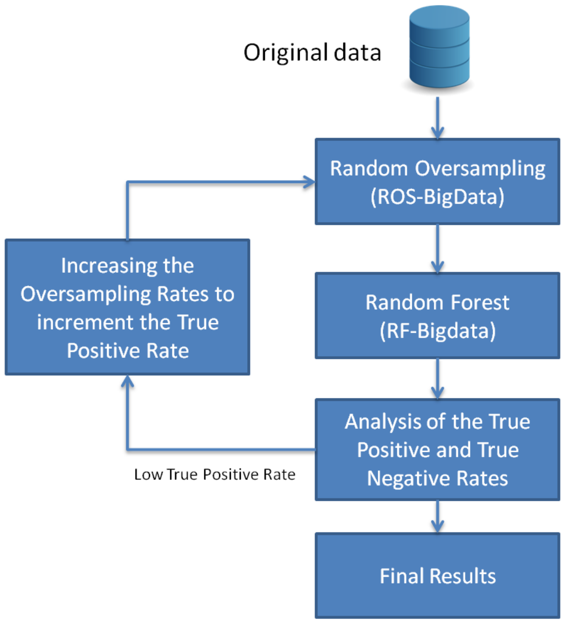
Original data
Random Oversampling (ROS-BigData)
Random Forest (RF-Bigdata)
Analysis of the True Positive and True Negative Rates
Final Results
Low True Positive Rate
Increasing the Oversampling Rates to increment the True Positive Rate

Fig. 2. Flowchart illustrating the process to increase the True Positive Rate

TABLE III. RESULTS OBTAINED WITH DIFFERENT OVERSAMPLING RATES USING 64 AND 192 MAPS AND 10 INTERNAL FEATURES FOR RF-BIGDATA

	64 $maps$		
$OversamplingRate$	TP_{rate}	TN_{rate}	$TP_{rate} \cdot TN_{rate}$
105%	0.619446	0.800734	0.496012
115%	0.653289	0.772620	0.504744
130%	0.704546	0.725117	**0.510878**
140%	0.704710	0.720721	0.507900
150%	0.722310	0.706574	0.510365
	192 $maps$		
$OversamplingRate$	TP_{rate}	TN_{rate}	$TP_{rate} \cdot TN_{rate}$
105%	0.642762	0.774510	0.497826
115%	0.681991	0.736557	0.502326
130%	0.733803	0.685623	**0.503113**
140%	0.734482	0.684857	0.503015
150%	0.765323	0.649534	0.497103

- As the value of TP_{rate} increases, the value of TN_{rate} decreases, therefore, is necessary to find out a balance between them in order to obtain the maximum precision in classification ($TP_{rate} \cdot TN_{rate}$). In this case, we have found a balance in the performance of both classes when an oversampling rate of 130% is used for both 64 and 192 maps.

In Figures 3 and 4 we show how the values of TP_{rate} and TN_{rate} vary depending on the oversampling rate using 64 and 192 maps, respectively.

At this point, we want to compare the best result obtained up to this point with the best results achieved by the second and third place in the ECBDL'14 Big Data Competition. Table IV presents these results.

Our best result obtained up to this point is not too far from the results obtained by participants who achieved the second and third place in the competition. Furthermore, we

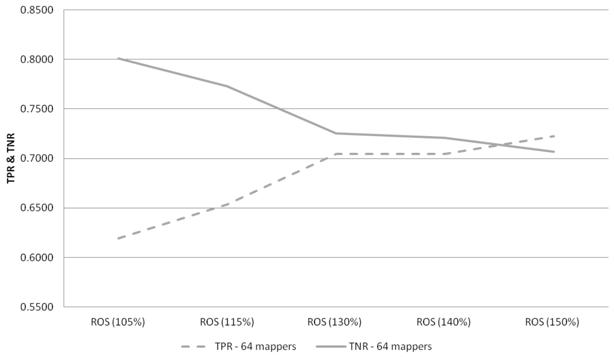

Fig. 3. TP_{rate} and TN_{rate} for increasing oversampling rate (64 maps)

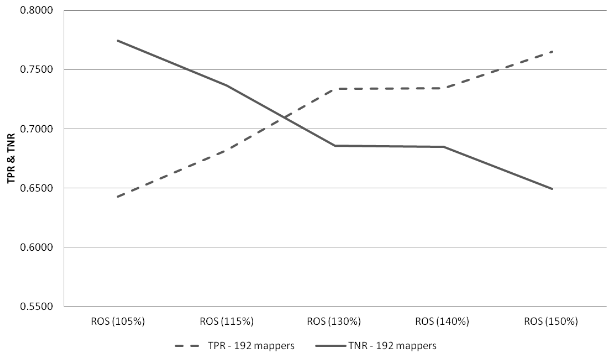

Fig. 4. TP_{rate} and TN_{rate} for increasing oversampling rate (192 maps)

TABLE IV. COMPARISON WITH THE SECOND AND THIRD PLACE IN ECBDL'14 BIG DATA COMPETITION

$Algorithm/Team$	TP_{rate}	TN_{rate}	$TP_{rate} \cdot TN_{rate}$
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

can see how our best result obtained using ROS-BigData with an oversampling rate of 130% and 64 maps is positioned above the result obtained by the third-placed.

D. Combining Random Oversampling with Higher Oversampling Ratios and Evolutionary Featuring Weighting

Since the dataset used in the ECBDL'14 Big Data Competition contains a fairly large number of features, we decided to use a new preprocessing component to improve the classification performance by obtaining the most relevant features. To do this, we use the DEFW-BigData algorithm, which calculates the importance of the features in terms of weights.

The DEFW-BigData algorithm generates a weight vector from which we select the features with the highest weights by setting the most appropriate threshold. We have selected this threshold from the results obtained in preliminary experiments which are not reported in this work. At the end of this process, we obtained a subset of 90 of the 631 original features.

Once we selected the features, we repeated the experiments using the ROS-BigData algorithm with different levels of oversampling rates from 100% to 150%. We also increased the number of internal features used by RF-BigData from 10 to 25 in order to further increase the overall precision. In Table V we present the results in test obtained with the subset of 90 features and 64 maps. This number of partitions is used because it led to the best results in the experiments of the previous section (see Table III). The value highlighted in bold corresponds to the best result.

TABLE V. RESULTS OBTAINED WITH FEATURE WEIGHTING AND DIFFERENT OVERSAMPLING RATES USING 64 MAPS AND 25 INTERNAL FEATURES FOR RF-BIGDATA

	64 $maps$		
$OversamplingRate$	TP_{rate}	TN_{rate}	$TP_{rate} \cdot TN_{rate}$
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**

From the results of the previous table we can extract the following conclusions:

- The use of a smaller subset of features has allowed us to obtain a greater accuracy compared to the results of the previous section.
- The DEFW-BigData algorithm has allowed to increase the TP_{rate} values but also the TN_{rate} values, producing an imbalance in the precision obtained in both classes. Please note that in the previous section we obtained a balance in the performance of both classes with an oversampling rate of 130% (see Table III).

In Figure 5 we show how the values of TP_{rate} and TN_{rate} vary depending on the oversampling rate.

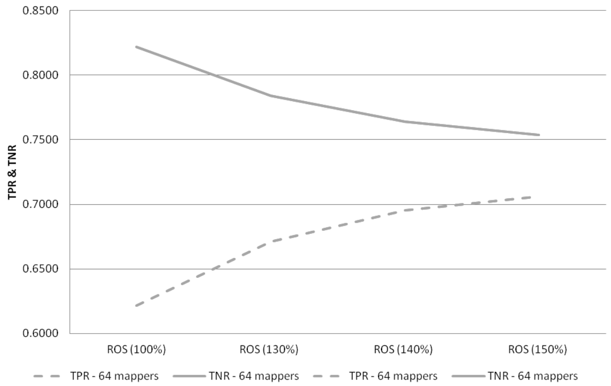

Fig. 5. TP_{rate} and TN_{rate} for increasing oversampling rate (64 maps)

The use of DEFW-BigData has allowed to obtain better performance results but has also caused the appearance of high differences between the precision obtained in the majority class and the less representative class. For this reason, we considered the conclusions reached in the previous section and we have decided to further increase the oversampling rate in order to find a balance between both classes and increment the overall precision. In Table VI we present the results in test obtained with the subset of 90 features, higher oversampling rates and 64 maps. We have highlighted in bold the value corresponding to best result.

From these results we conclude that it is necessary the use of a high oversampling rates in order to find a balance in

TABLE VI. RESULTS OBTAINED WITH FEATURE WEIGHTING AND HIGHER OVERSAMPLING RATES USING 64 MAPS AND 25 INTERNAL FEATURES FOR RF-BIGDATA

	64 $maps$		
$OversamplingRate$	TP_{rate}	TN_{rate}	$TP_{rate} \cdot TN_{rate}$
160%	0.718692	0.741976	0.533252
170%	0.730432	0.730183	**0.533349**
180%	0.737381	0.722583	0.532819

the performance of both classes. In this case, we have found this balance when an oversampling rate of 170% is used. The average number of copies per instance of the minority class is 81.6 $(48 \cdot 1.7)$ and, therefore, an important representation of this class is obtained.

In Figure 6 we show how the TP_{rate} and TN_{rate} vary depending on the oversampling rate.

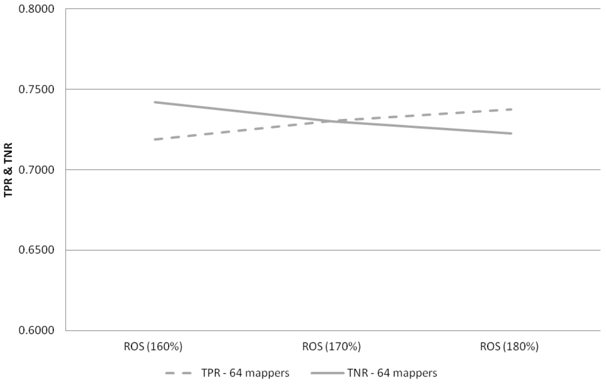

Fig. 6. TP_{rate} and TN_{rate} for increasing oversampling rate (64 maps)

Finally, in Table VII we show the best results in test achieved from the top three participants in the ECBDL'14 Big Data Competition.

TABLE VII. RESULTS ACHIEVED FROM THE TOP THREE PARTICIPANTS IN THE ECBDL'14 BIG DATA COMPETITION

	64 $maps$		
$Algorithm/Team$	TP_{rate}	TN_{rate}	$TP_{rate} \cdot TN_{rate}$
Efdamis (1°)	0.730432	0.730183	**0.533349**
ICOS (2°)	0.703210	0.730155	0.513452
UNSW (3°)	0.699159	0.727631	0.508730

IV. CONCLUSION

In imbalanced classification problems the lack of density of the minority class causes a negative impact in the performance. In big data, the impact is further increased when the original data are partitioned into subsets by the MapReduce procedure. For this reason, our hypothesis stated that an increment in the density of the underrepresented class by using higher oversampling ratios could improve the classification performance.

The experiments carried out over the ECBDL'14 Big Data Competition dataset support this hypothesis and have yielded an improvement in the overall accuracy. Setting the oversampling ratio to a value that balances the TP_{rate} and TN_{rate} values leads to the best performance in terms of accuracy.

ACKNOWLEDGMENT

This work was partially supported by the Spanish Ministry of Economy and Competitiveness under projects TIN2014-57251-P, TIN2013-47210-P and the Andalusian Research Plans P12-TIC-2958, P11-TIC-7765 and P10-TIC-6858.

REFERENCES

[1] P. Zikopoulos, C. Eaton, D. DeRoos, T. Deutsch and G. Lapis, "Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data," *McGraw-Hill*, 2011.

[2] S. Madden, "From Databases to Big Data," *IEEE Internet Computing*, vol. 16, no. 3, pp. 4–6, 2012.

[3] X. Wu, X. Zhu, G. Q. Wu and W. Ding, "Data Mining with Big Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, 2014.

[4] A. Fernández, S. Río, V. López, A. Bawakid, M. J. del Jesus, J. M. Benítez and F. Herrera, "Big Data with Cloud Computing: An Insight on the Computing Environment, MapReduce and Programming Frameworks,"*WIREs Data Mining and Knowledge Discovery*, vol. 4, no. 5, pp. 380–409, 2014.

[5] H. He and E. A. García, "Learning from imbalanced data,"*IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[6] V. López, A. Fernández, S. García, V. Palade and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113–141, 2013.

[7] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters,"*Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.
IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1388–1400, 2010.

[8] G. M. Weiss, F. J. Provost, "Learning when training data are costly: the effect of class distribution on tree induction," *Journal of Artificial Intelligence Research*, vol. 19, pp. 315–354, 2003.

[9] G. M. Weiss, "Mining with rarity: a unifying framework," *SIGKDD Explorations*, vol. 6, no. 1, pp. 7–19, 2004.

[10] S. Río, V. López, J. M. Benítez and F. Herrera, "On the use of MapReduce for imbalanced big data using Random Forest,"*Information Sciences*, vol. 285, no. 0, pp. 112–137, 2014.

[11] "ECBDL'14 Big Data Competition," *[Online; accessed March 2015]* (http://cruncher.ncl.ac.uk/bdcomp/), 2014.

[12] I. Triguero, S. Río, V. López, J. Bacardit, J. M. Benítez and F. Herrera, "ROSEFW-RF: The winner algorithm for the ECBDL'14 Big Data Competition: An extremely imbalanced big data bioinformatics problem,"*Knowledge-Based Systems*, in press 2015.

[13] "Apache Mahout Project," *[Online; accessed March 2015]* (http://mahout.apache.org/), 2013.

[14] T. White, *Hadoop, The Definitive Guide.* O'Reilly Media, Inc., 2012.

[15] V. García, R. A. Mollineda, J. S. Sánchez, "On the k-NN performance in a challenging scenario of imbalance and overlapping," *Pattern Analysis and Applications*, vol. 11, no. 3–4, pp. 269–280, 2008.

[16] V. López, A. Fernández J. G. Moreno-Torres and F. Herrera, "Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6585–6608, 2012.

[17] J. Bacardit, P. Widera, A. Marquez-Chamorro, F. Divina, J. S. Aguilar-Ruiz and N. Krasnogor, "Contact map prediction using a large-scale ensemble of rule sets and the fusion of multiple predicted structural features," *Bioinformatics*, vol. 28, no. 19, pp. 2441–2448, 2012.