Neural Process Lett (2015) 42:603–617
DOI 10.1007/s11063-014-9376-3

An Efficient Over-sampling Approach Based on Mean Square Error Back-propagation for Dealing with the Multi-class Imbalance Problem

R. Alejo · V. García · J. H. Pacheco-Sánchez

Published online: 17 August 2014

Abstract In this paper a new dynamic over-sampling method is proposed, it is a hybrid method that combines a well known over-sampling technique (SMOTE) with the sequential back-propagation algorithm. The method is based on the back-propagation mean square error (MSE) for automatically identifying the over-sampling rate, i.e., it allows only the use of necessary training samples for dealing with the class imbalance problem and avoiding to increase excessively the (neural networks) NN training time. The main aim of the proposed method is to obtain a trade-off between NN classification performance and NN training time on scenarios where the training data set represents a multi-class classification problem, it is high imbalanced and it might request a large NN training time. Experimental results on fifteen multi-class imbalanced data sets show that the proposed method is promising.

Keywords High multi-class imbalance · Sequential back-propagation algorithm · Mean square error · Dynamic over-sampling technique · SMOTE

1 Introduction

Back-propagation is now the most widely used tool in the field of artificial neural networks (NN). However, despite the general success of the back-propagation, several major deficien-

R. Alejo (✉)
Tecnológico de Estudios Superiores de Jocotitlán, Carretera Toluca-Atlacomulco KM. 44.8, Ejido de San Juan y San Agustín, 50700 Jocotitlán, Mexico
e-mail: ralejoll@hotmail.com

V. García
Department of Electrical and Computer Engineering, Instituto de Ingeniería y Tecnología,, Universidad Autónoma de Ciudad Juárez, Av. del Charro 450 Norte, 32310 Ciudad Juárez, Chihuahua, Mexico
e-mail: vgarciaj@gmail.com

J.H. Pacheco-Sánchez
Instituto Tecnológico de Toluca, Av. Tecnológico s/n Ex-Rancho La Virgen, 52140 Metepec, Mexico
e-mail: hpacheco@ittoluca.edu.mx

cies are still needed to be solved. The major disadvantage of back-propagation is the slow rate of convergence of net output error [27,38]. This is especially difficult in class imbalance problems [3,35], and often it is the cause of the poor classifications performance of the NN.

The class imbalance problem occurs when, in a classification problem, there are many more samples of some classes than others [13]. This problem exists in many real-world domains, such as spotting unreliable telecommunications customer, detection of oil spills in satellite radar images, detection of fraudulent telephone calls, information retrieval and filtering task and so on [24].

Much research has been done in addressing the class imbalance problem [21,39]. In the back-propagation in "batch mode" [19], it is very popular the use of cost function to deal with class imbalance problem (e.g. see Ref. [2,6,25,26,31,35]). In these approaches, the basic idea is modify the error function of the back-propagation by introducing different costs associated with making errors in different classes, for dealing class imbalance.

In the "sequential" back-propagation (which estimates the error based on individual error training sample [19]), a common practice is to apply re-sampling techniques on the original training dataset, either by over-sampling or under-sampling or both. The re-sampling methods are the most researched because they are independent of the underlying classifier and can be easily implemented for any problem [32].

The simplest method to increase the size of the minority class corresponds to random over-sampling, which is a non-heuristic method that balances the class distribution through the random replication of positive examples [21,23]. Nevertheless, since this method replicates existing examples in the minority class, overfitting is more likely to occur.

Others over-sampling methods with some heuristic techniques have been proposed. Chawla et al. [7] proposes the synthetic minority over-sampling technique (SMOTE), which generates new synthetic minority samples by interpolating between several preexisting positive examples that lie close together. In Ref. [17] the Borderline-SMOTE was presented, which, it only over-samples the borderline samples of the minority class. Adaptive synthetic sampling (ADASYN) was proposed as a technique that uses a systematic method for adaptively creating different amounts of synthetic data according to their distributions [20]. García et al. [16] uses surrounding neighborhood approaches with the aim of generating artificial minority examples, but taking both the proximity and the spatial distribution of the examples into account.

On the other hand, random under-sampling is the most popular technique among this nature, whose aim is at balancing the dataset through the random removal of negative examples. Despite its simplicity, it has empirically been shown to be one of the most effective re-sampling methods [23]. However, some works agree [9,21,36] in that the random under-sampling is weakened in multi-class scenarios and it can cause great performance reduction to those majority classes. Many other under-sampling proposals are based on a more intelligent selection of the negative examples to be eliminated [21], for example the Tomek link, the nearest neighbor rule (NNR), the condensed NNR [4], the Gabriel Graphs [2] genetic algorithms [15], so on. It is also common to blend the over and under samplings methods [2,4,8,10]

As been stated by several authors, the re-sampling methods may entail important criticism and limitations:

1. How to automatically discover the proper amount of sampling (sampling rate)? [8].
2. In severe class imbalance problems, over-sampling methods modifies the data set probability distribution, cause longer training time and suffers from high computational cost in terms of memory [9,13].

3. The under-sampling techniques involves a lost of information which can be detrimental for the classifier performance [9,21,36].

The present paper focus in the first and second point with the aim to determine a proper over-sampling rate without sacrificing the performance on the minority classes, while also reducing the training time and the computational cost. In brief, we propose a dynamic method that allows the efficient use of an over-sampling strategy on severe multi-class imbalanced problems. The method is based on the back-propagation mean square error (MSE) for automatically identifying the over-sampling rate. More specifically, the main contributions of this paper/method/technique are :

1. To deal with severe multi-class imbalance problems, which have been less investigated [36].
2. To provide an efficient way of over-sampling minority classes on highly multi-class imbalance problems.
3. A simple method for automatically finding the over-sampling rate which does not need the free parameters and it is very easy to implement.

The rest of this paper is organized as follows. Related works are briefly reviewed in Sect. 2. In Sect. 3 we introduce the proposed method for tackling the multi-class imbalance problem, and the Sects. 4 and 5 show the experimental set up and results, respectively. Finally, Sect. 6 is for concluding remarks.

2 Related Works

A common practice for dealing with class imbalanced data sets is to re-balance them artificially through the re-sampling techniques [21]. However, a supported concern in researches in data mining and machine learning has been to deal with improving the re-sampling methods [8,34]. Some efforts have been addressed to overcome one of its main criticisms: *to find the proper over or under sampling rate*.

Fernández Navarro et al. [13,14] present a dynamic over-sampling algorithm to deal with multi-class imbalance problems on Radial Basis Function and Multilayer Perceptron, respectively. In that approach, the data set is modified into two stages. In the first stage, the data set is preprocessed with SMOTE to reduce the class imbalance ratio. The number of samples created by SMOTE in this stage is less than $1/2 * J$ where J is the number of classes. In the second stage, a memetic Algorithm [29] is proposed to obtain the best parameters for NN. Next, the SMOTE algorithm is applied to the minimum sensitivity class to decrease the imbalance problem. This stage is in order to run while stop condition is not succeed. However, the authors do not present the class imbalance ratio resulting from finishing this proccess or its computational efficiency.

Chawla et al. [8] propose a wrapper paradigm that discovers the amount of re-sampling for a data set based on optimizing evaluation functions like the f-measure, area under the ROC curve (AUROC), cost, cost-curves, and the cost dependent f-measure. The classifiers base were C4.5 and RIPPER. To discover the re-sampling amounts, the five-cross validation method is applied at the training set, and the next two stages are to run. First, the wrapper finds the under-sampling percentages for the dataset. The process consists in decreasing the majority class to improve the minority classification performance without sacrificing performance on the majority class. In the second stage, the amount of SMOTE is incremented, and it is evaluated whether the performance is increased with the new SMOTE amount.

This process repeats, greedily, until not performance gains are observed. Once that the re-sampling amounts are obtained, they are used for re-sampling the original training dataset and the classifier is trained again. The result presented by the authors demonstrated the effectiveness of the generalization performance of the proposed method. However, the over-sampling amounts showed in some datasets are much higher than to balance at 100 % the dataset, therefore, we considered that the wrapper paradigm might be not efficient in NN. In this respect, a very similar work is presented by Debowski et al. [10], but at difference of Chawla et al. [8], it shows important weaknesses. For example, the stopping condition is not clear and we consider that the experimental framework is limited.

Ref. [30,33] present the snowball method to deal with the class imbalance problem. It is a dynamic over-sampling training method for NN. The basic idea is to first train the NN only with the examples of the minority class. Next, they use a dynamic training which includes all examples of minority class and a gradual increasing of number of examples of the majority class in the training. In this way, the effect of undoing the presentation of minority class examples can be greatly reduced. However the authors contradict their owns results, in Ref. [30] Murphey says, that on Back-propagation, the classification accuracy increased over the minority class with the price of dramatically decreasing classification accuracy over the majority class, and Ou et al. [33] show that Snowball method gives the best performance without any loss on the majority classes.

A similar work is presented by Bo-Yu Li [28]. The basic idea is for training the NN with a dynamic threshold learning algorithm. This method uses multiple dynamic threshold parameter to gradually remove some training samples that can be classified correctly by the NN, and, in this way to get a class balance and to improve the classification performance over the minority classes. But in the same way that in others works [2,30], the cost of improving the minority classes performance is to sacrifice the effectiveness of the NN on the majority classes.

We proposed a method which is different to the other works in the “approach”, i.e., it deals to reduce as much as is possible the NN training time, when it is trained with multi-class imbalanced datasets, without significantly loss of the classification performance, and in this way to deal with one of the major disadvantages of back-propagation (the NN training time). This is very important on scenarios where the NN is trained from datasets of considerable size with highly imbalanced classes. For example in the classification of hyperspectral remote sensing images, where the over-sampling strategies may increase too much the training time or the under-sampling might seem inapplicable due to the important loss of information.

3 Proposed Method for Dealing with Multi-class Imbalance Problems

It is well known that in the back-propagation algorithm the class imbalance problem generates unequal contributions to the mean square error (MSE) in the training phase [2,3]. So the training process also becomes slow and it takes long time to converge to the expected solution.

The main problem consists in that the majority classes produce the major contribution to the MSE. Let us consider the next: Given a training dataset (TDS) with two classes ($J = 2$) such that $Q = \sum_j^J Q_j$ and Q_j is the number of samples from class j, and supposing that the MSE by class can be expressed as

$$E_j(U) = \frac{1}{Q} \sum_{i=1}^{Q_j} \sum_{p=1}^{J} (t_p^i - z_p^i)^2 , \tag{1}$$

where t_p^i is the desired output and z_p^i is the actual output of the network for the sample i. Then the overall MSE can be expressed as

$$E(U) = \sum_{j=1}^{J} E_j(U) = E_1(U) + E_2(U)\,. \tag{2}$$

If $Q_1 << Q_2$ then $E_1(U) << E_2(U)$ and $\|\nabla E_1(U)\| << \|\nabla E_2(U)\|$, where the operator ∇ denotes the gradient of the error function. Consequently, $\nabla E(U) \approx \nabla E_2(U)$. So, $-\nabla E(U)$ is not always the best direction to minimize the MSE in both classes [3].

In the batch mode back-propagation is a common practice to balance the MSE including a cost function $\gamma(j)$ for balancing the MSE, i.e, $\gamma(1)\|\nabla E_1(U)\| \approx \gamma(2)\|\nabla E_2(U)\|$ [2,6,25,26,31,35]. Nevertheless, in the sequential mode of back-propagation it is not trivial task. For this reason is normal the use of the re-sampling techniques to deal with balancing the MSE on the training process.

In this work we propose a dynamic over-sampling technique to balance the MSE on the training stage when a multi-class imbalanced dataset is used. The propose method consists in two steps:

1. *Before training*: The TDS is balanced at 100 % through of an effective over-sampling technique. In this work we use SMOTE [7].
2. *During training*: The MSE by class (E_j) is used to determinate the number of samples by class (or ratio of class) in order to forward it to the NN. The equation employed to obtain the ratio of class is defined as

$$ratio_j = \frac{E_{max}}{E_j} * \frac{Q_j}{Q_{max}}; \quad for\ j = 1, 2, ..., J\,, \tag{3}$$

   where J is the number of classes in the dataset and max identifies at the largest majority class. The Eq. 3 allows to balance the MSE by class reducing the impact of the class imbalance problem on the NN. The Algorithm 1 shows the implementation of this step.

The main peculiarity of our proposed method is that in the training stage only uses the necessary samples for dealing with the class imbalance problem and in this way to avoid a poor performance of classifications resulting from the NN over the minority classes and the NN training time is not increased excessively.

The proposed method (detailed in Algorithm 1) shows the next advantages:

1. It is a simple method with a single classifier.
2. It does not need more free parameters than the standard back-propagation.
3. It only uses the necessary samples in the training stage to get a MSE by class relatively balanced.

4 Experimental Setup

In order to evaluate the validity and performance of the technique just proposed, we have accomplished thorough experiments on several multi-class imbalanced data sets. In this section, we will describe the techniques, data sets and experimental framework used in the paper.

Algorithm 1 MSE back-propagation over-sampling technique (MSEBPOS).

Input: N (number of input nodes), M (number of middle neurodes), J (classes), $\mathbf{x}^{(q)}$ (the exemplar vectors), $\mathbf{t}^{k(q)}$ (the paired identifier vectors), I number of epochs; and learning rate η.

Output: the weights $\mathbf{w} = (w_{11}, w_{21}, ..., w_{NM})$ $\mathbf{u} = (u_{11}, u_{21}, ..., w_{MJ})$, the total and partial MSE (E, E_j) respectively.

```
INIT( ):
Read MLP file ($N, M, J, Q, I$ and $\eta$);
Generate initial weights randomly between $-0.5$ and $0.5$;
Initial $ratio_j = Q_j/Q_{max}$; for $j = 1, 2, ..., J$
LEARNING( ):
while $i < I$ or $E > 0.001$ do
  for $q = 0$ to $Q$ do
    if Random( ) $<= ratio_{class(\mathbf{x}^q)}$ then
      Forward($\mathbf{x}^q$);
      Update($\mathbf{x}^q$);
    end if
  end for
  $ratio_j = (E_{max}/E_j) * Q_j/Q_{max}$; for $j = 1, 2, ..., J$
end while
FORWARD($\mathbf{x}^q$):
for $m = 0$ to $m < M$ do
  for $n = 0$ to $n < N$ do
    $y_m \leftarrow y_m + x_n^q * w_{nm}$;
  end for
  $y_m = net(y_m)$;
end for
for $j = 0$ to $j < J$ do
  for $m = 0$ to $m < M$ do
    $z_j \leftarrow z_j + u_{mj} * y_m$;
  end for
  $z_j \leftarrow net(z_j)$;
end for
UPDATE($\mathbf{x}^q$):
for $m = 1$ to $M$ do
  for $j = 1$ to $J$ do
    $u_{mj}^{r+1} \leftarrow u_{mj}^r + \eta\{(t_j^{(q)} - z_j^{(q)})[z_j^{(q)}(1 - z_j^{(q)})]y_m^{(q)}\}$;
  end for
  for $n = 1$ to $N$ do
    $w_{nm}^{r+1} \leftarrow w_{nm}^r + \eta\{\sum_{j=1,J}(t_j^{(q)} - z_j^{(q)})[z_j^{(q)}(1 - z_j^{(q)})]u_{mj}^{(r)}\}x_n[y_m^{(q)}(1 - y_m^{(q)})][x_n^{(q)}]$;
  end for
end for
```

4.1 Re-sampling Methods

The class imbalance problem has been addressed by re-sampling techniques, which artificially balance the original data set, either by over-sampling of the minority class or under-sampling of the majority class or both. In this work, we have used a renowned over-sampling technique called SMOTE proposed by Chawla et al. [7]. This method generates artificial examples of the minority class by interpolating existing instances that lie close together. For each minority sample, it finds the k intra-class nearest neighbors, and then synthetic samples are generated in the direction of some or all of those nearest neighbors. As reported in paper by Chawla et al. [7], in our experiments the k value has been restricted to five nearest neighbors, bearing in mind that the aim of the present study is not at finding the optimal k value. Besides, a constant k value allows to make easier the interpretation of results focused on our proposal.

In addition, we decided to add examples until a balanced distribution was reached. This decision was intentioned by two aims: (a) simplicity (to avoid use of many free parameters) and (b) effectiveness. Results obtained with the other classifiers [37], have shown that when AUC is used as a performance measure, the best class distribution for learning tends to be near the balanced class distribution.

Although the class imbalance problem has been claimed as the main factor that significant degrades the performance of classifiers. Several studies have pointed out that the degradation is also related to other factors such as small disjuncts, high dimensionality and class overlapping [5]. In order to handle both class imbalance and class overlapping, we have jointly used an over-sampling and a data cleaning method with the double aim of balancing the skewed classes and removing any erroneous and harmful majority example. Specifically, in this work we have chosen the aforementioned SMOTE and the Gabriel graph editing (GGE) techniques, which have been shown to be suitable to deal with the two issues for the back-propagation learning procedure [2].

4.2 Description of the Experimental Data Sets

Five real-world remote sensing data sets were selected to test our proposal: MSE back-propagation over-sampling (MSEBPOS) technique. The Cayo data set comes from a particular region in the gulf of Mexico [2]. The Feltwell data set represents an agricultural area near the village of Fetwell (UK) [6]. The Satimage and Segment data sets are from the UCI Machine Learning Database Repository [1]. The 92AV3C dataset[1] corresponds to a hyperspectral image (145 × 145 pixels, 220 bands, 17 classes) taken over Northwestern Indianas Indian Pines by the AVIRIS sensor. In this work, we employed a reduced version of this data set with six classes (2, 3, 4, 6, 7 and 8) and 38 attributes as in [2].

As we are interested in analyzing the technique proposed on highly imbalanced multi-class data sets, each original data set was altered by combining and/or reducing the size of some classes in order to construct fifteen multi-class data sets with a diverse number of class distributions. Table 1 reports a summary of the original classes that were joined to shape the majority and minority classes. The third and fourth columns indicate the original and final classes, respectively. The number between parentheses represents the classes that were joined to shape the majorities classes. For example for the MCAA subset the classes 1, 3, 6, 7 and 10 from original database (Cayo) were joined to integrate its first majority class and the classes 8, 9 and 11 to ingrate its second majority class. So the result of this process is a subset (MCAA) from Cayo with five classes: two majorities and three minorities classes. The main difference between the subsets obtained from the original database is the classes that were integrated to shape their majorities classes, for example, the difference between MCAB and MCAC is that for MCAB the class 4 is part of the one of its majorities classes meanwhile that in MCAC the class 4 is a minority class.

The above described process was performed in all datasets used in this work (Cayo, Fetwell, Satimage, Segment and 92AV3C) and to further reduce the minorities classes random size under sampling was employed. The main characteristics of the new produced benchmarking data sets are showed in Table 2.

4.3 Experimental Framework

An empirical comparison between the MSE back-propagation over-sampling (MSEBPOS) technique here proposed and other re-sampling strategies were performed over a total of fifteen data sets by using the multi-layer perceptron (MLP) neural network trained with the sequential back-propagation (SBP) algorithm. A stratified ten-fold cross validations was adopted for the present study. For each fold, nine parts were pooled as the training data, and the remaining block was employed as an independent test set. All the training sets were pre-processed by the MSEBPOS technique and the original SMOTE. Apart from these methods,

[1] https://engineering.purdue.edu/biehl/MultiSpec/hyperspectral.html

Table 1 Detail of the classes merge process to get highly imbalances data sets

Original data set	Final data set	Original classes	Final classes	Classes joined						
				1	2	3	4	5	6	7
Cayo	MCAA	11	5	(1, 3, 6, 7, 10)	2	(8, 9, 11)	4	5	–	–
	MCAB		4	(1, 3, 4, 6, 7, 10)	2	(8, 9, 11)	5	–	–	–
	MCAC		4	(1, 3, 5, 6, 7, 10)	2	(8, 9, 11)	4	–	–	–
Feltwell	MFEA	5	5	1	2	3	4	5	–	–
	MFEB		4	(1, 2)	3	4	5	–	–	–
	MFEC		4	(1, 4)	2	3	5	–	–	–
Satimage	MSAA	6	6	1	2	3	4	5	6	–
	MSAB		5	(1, 2)	3	4	5	6	–	–
	MSAC		5	(1, 4)	2	3	5	6	–	–
Segment	MSEA	7	7	1	2	3	4	5	6	7
	MSEB		6	(1,3)	2	4	5	6	7	–
	MSEC		6	(1,7)	2	3	4	5	6	–
92AV3C	M92A	16	6	1	2	3	4	5	6	–
	M92B		5	1	2	3	(4, 5)	6	–	–
	M92C		5	1	2	(3, 4)	5	6	–	–

Table 2 A brief summary of some characteristics of the data sets used in the experimental stage

Data set	#Ex.	#Attr.	#Examples per class						
			1	2	3	4	5	6	7
MCAA	6019	4	2941	293	2283	369	133	–	–
MCAB			3310	293	2283	133	–	–	–
MCAC			3074	293	2283	369	–	–	–
MFEA	8536	15	3531	2441	91	2295	178	–	–
MFEB			5972	178	91	2295	–	–	–
MFEC			5826	2441	91	178	–	–	–
MSAA	4697	36	1508	1533	104	1358	93	101	–
MSAB			3041	101	104	1358	93	–	–
MSAC			2866	1533	104	101	93	–	–
MSEA	1470	19	330	50	330	330	50	50	330
MSEB			660	50	330	330	50	50	–
MSEC			660	50	330	330	50	50	–
M92A	5063	38	190	117	1434	2468	747	106	–
M92B			190	117	1434	3215	106	–	–
M92C			190	117	3902	106	747	–	–

other two variants in combination with the GGE were included in the study: MSEBPOS+GGE and SMOTE+GGE. Here, it is worth mentioning that all the original and resampled data sets by SMOTE and SMOTE+GEE were used to build the prediction model with the non-modified SBP algorithm.

In the training process, for both the SBP and MSEBPOS, the weights were randomly initialized ten times. Therefore, the results from classifying the test samples were averaged between the ten runs and the ten different initialization weights. The learning rate (η) was set to 0.1 and the stopping criterion was established at 5,000 epoch or the MSE value is lower than 0.001. A single hidden layer was used, where for each data set the number of neurons was obtained by a trial and error strategy: Cayo = 7, Feltwell = 6, Satimage = 12, Segment = 10 and 92AV3C = 10.

4.4 Performance Evaluation

Several empirical and theoretical studies have shown that the plain accuracy and/or error rates are strongly biased with respect to data imbalance, which might produce misleading conclusions [21,31]. In order to face this shortcoming, alternative performance evaluation measures have been proposed. One of the most widely-used graphical evaluation methods is the receiver operating characteristic (ROC) curve, which is a tool for visualizing, organizing and selecting binary classifiers based on their trade-offs between true positive rates and false positive rates [12]. A quantitative representation is the Area Under Curve (AUC) ROC, which "summarizes" the quality of the classifier. In problems where the classes can be more than two, the AUC can be defined as [18]:

$$AUC = \frac{2}{\|J\|(\|J\|-1)} \sum_{j_i, j_k \epsilon J} AUC_R(j_i, j_k) \ , \tag{4}$$

where $\text{AUC}_R(j_i, j_k)$ is the area under the curve for each pair of classes j_i and j_k.

4.5 Criteria for Evaluating Experimental Results

In this work, we have employed the Friedman test for evaluating the experimental results with the aim of verifying the hypothesis of improved performance of the re-sampling techniques here used. It is a non-parametric statistical test that performs multiple comparisons among the algorithms considered over a collection of data sets [11]. The procedure starts by computing the ranks of the algorithms or strategies for each dataset separately, where the best performing algorithm gets the rank of 1, the second best rank 2, and so on. In case of ties, average ranks are computed. Let r_i be the rank of the $j-$th of K algorithms on the $i-$th of N data sets. The next step is in order to obtain the average ranking for each algorithm, $R_j = \frac{1}{N}\sum_i r_i^j$. Under the null hypothesis which states that all algorithms behave similarly and therefore their ranks R_j should be equal, the Friedman statistic can be computed as follows:

$$\chi_F^2 = \frac{12N}{K(K+1)}\left(\sum_j R_j^2 - \frac{K(K+1)^2}{4}\right) \ . \tag{5}$$

The χ_F^2 is distributed according to the Chi-square distribution with $K-1$ degrees of freedom, when N and K are big enough. Owing to χ_F^2 presents an undesirable conservative behavior, Iman and Davenport [22] have devised a better statistic distributed according to the $F-$distribution with $K-1$ and $(K-1)(N-1)$ degrees of freedom,

$$F_F = \frac{(N-1)\chi_F^2}{N(K-1)-\chi_F^2} \ . \tag{6}$$

If the null-hypothesis is rejected, we can use a Bonferroni-Dun post-hoc test, which compares a control algorithm with the $K-1$ algorithms. The performance of two algorithms is significantly different if the corresponding average ranks is at least as great as its critical

Table 3 Classification performance on fifteen data sets measured using AUC and average rank (AR)

Dataset	MSEBPOS+GGE	SMOTE+GGE[1]	MSEBPOS	Imbalanced[1]	SMOTE[1]
MCAA	0.897	0.906	0.847	0.739	0.904
MCAB	0.927	0.932	0.893	0.763	0.933
MCAC	0.907	0.911	0.863	0.754	0.910
MFEA	0.909	0.907	0.913	0.802	0.930
MFEB	0.916	0.926	0.899	0.766	0.941
MFEC	0.835	0.851	0.906	0.807	0.931
MSAA	0.764	0.765	0.832	0.751	0.834
MSAB	0.788	0.790	0.806	0.698	0.837
MSAC	0.816	0.821	0.811	0.716	0.847
MSEA	0.912	0.913	0.945	0.930	0.944
MSEB	0.931	0.926	0.945	0.916	0.947
MSEC	0.905	0.897	0.937	0.921	0.934
M92A	0.758	0.775	0.802	0.793	0.833
M92B	0.736	0.750	0.785	0.772	0.849
M92C	0.844	0.874	0.845	0.853	0.887
Average	0.856	0.863	0.869	0.799	0.897
Average Rank	3.8	2.93	2.67	4.33	1.27

[1] Classification using SBP

difference,

$$CD = q_\alpha \sqrt{\frac{K(K+1)}{6N}}\ , \tag{7}$$

where the q_α values is based on the studentized range statistic divided by $\sqrt{2}$ (Table 5(b) in [11]).

5 Results and Discussion

Table 3 reports the detailed AUC results of each problem and the average AUC values across all databases using the different strategies here explored: the imbalanced case, the MSE back-propagation over-sampling technique (MSEBPOS), SMOTE, MSEBPOS+GGE and SMOTE+GGE. The average ranks (Friedman score) are also given. As expected, classification with the imbalanced data set (the non-preprocessed training set) yields the poorest AUC value. The two best performing algorithms corresponds to SMOTE and MSEBPOS techniques. This can be further confirmed by noting the average ranks, which provides a useful comparison of the algorithms, where the imbalanced case has the highest average rank. The SMOTE and MSEBPOS were ranked with 1.27 and 2.67, respectively.

In order to detect whether there exist statistical differences between the AUC results of the techniques studied, we employed the Iman-Davenport statistic. This computation produced $F_F = 17.3745$, distributed according to F distribution with 4 and 56 degrees of freedom. The $p-$value returned by using $F(4, 56)$ was $25.41E - 10$. As the $p-$value is lower than a significant level of $\alpha = 0.05$, the null hypothesis which states that all algorithms here explored behave equally can be rejected. Hence, we carried out a post-hoc statistical analysis

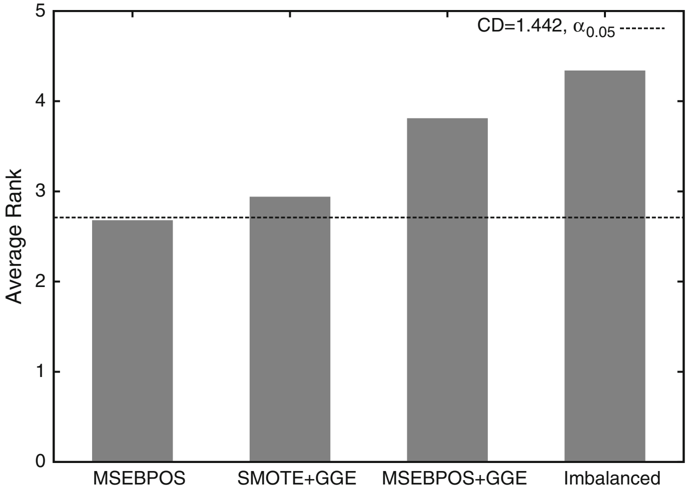

Fig. 1 Bonferroni-Dunn graphic for AUC

by using Bonferroni-Dunn procedure to compare each strategy against the control classifier, which corresponds to the best strategy.

Figure 1 plots the average ranks for each strategy, which appear sorted according to their ranks. The horizontal line represents the threshold of the critical difference value computed by the Bonferroni-Dunn test with $\alpha = 0.05$. This line is equal to the sum of the lowest rank (1.27) and the CD value (1.442). Those bar (algorithms) above this cut line perform significantly worse than the best model. Observing the results from Fig. 1, we can see that only the MSEBPOS behaves equally to SMOTE. The other three strategies do not perform significantly better. Therefore, the MSEBPOS appears to be a suitable and effective approach to deal with multi-class imbalance problems.

We have also analysed the final set size obtained by each algorithm. Table 4 reports the set size ratio on each database. It was computed as $ratio_k = Q_k/Q_{SMOTE}$, where Q_k and Q_{SMOTE} are the size of the preprocessed and balanced data sets (this by SMOTE), respectively. As can be observed, the re-sampling methods based on the MSE back-propagation over-sampling technique (MSEBPOS and MSEBPOS+GGE) have achieved less than 0.53 of set size ratio, what means significant saving in time computing (see Table 5) and storage requirements, when compared to the balanced data set by SMOTE. From Table 5, we also can see that the time processing rate was remarkable reduced more than twice SMOTE.

For the sake of a visual comparison and with the aim of analyzing the performance of a re-sampling approach in terms the AUC and the set size ratio, we have employed a scatter-plot of the size ratio versus the AUC, values by means of average ranks. Fig. 2 displays all the strategies studied (including the imbalanced data set), where the $x-$ axis are the average ranks of AUC results and the $y-$axis are the average ranks of the set size ratio. In such a way, points close to the origin (0, 0) of the plot might corresponds to the best methods with a good balanced trade-off between performance and size complexity. Similarly, we have plotted the average ranks of the time processing versus the AUC values in Fig. 3.

From Figs. 2 and 3, one can observe that the MSEBPOS approach lies the nearest from the origin of the plot, which suggest that this technique has the most suitable trade-off in terms of performance and size, as well as, time processing. Alternative, MSEBPOS+GGE

Table 4 The size ratio of each dataset obtained by the strategies studied. It was computed taking as reference at SMOTE: $ratio_k = Q_k/Q_{SMOTE}$, where Q_k and Q_{SMOTE} are the size of data set processed by the strategy k and SMOTE, respectively

Dataset	MSEBPOS+GGE	SMOTE+GGE[1]	MSEBPOS	Imbalanced[1]	SMOTE[1]
MCAA	0.36	0.65	0.48	0.41	1.00
MCAB	0.40	0.76	0.50	0.45	1.00
MCAC	0.40	0.68	0.56	0.49	1.00
MFEA	0.51	0.65	0.67	0.48	1.00
MFEB	0.30	0.69	0.46	0.36	1.00
MFEC	0.43	0.79	0.57	0.37	1.00
MSAA	0.31	0.33	0.69	0.51	1.00
MSAB	0.17	0.24	0.42	0.31	1.00
MSAC	0.15	0.15	0.40	0.33	1.00
MSEA	0.64	0.79	0.79	0.64	1.00
MSEB	0.32	0.60	0.49	0.37	1.00
MSEC	0.37	0.57	0.67	0.37	1.00
M92A	0.56	0.93	0.43	0.34	1.00
M92B	0.39	0.97	0.43	0.31	1.00
M92C	0.51	0.99	0.45	0.26	1.00
Average	0.39	0.65	0.53	0.40	1.00
Average Rank	1.53	3.40	3.23	1.83	5.00

[1] Classification using SBP

Table 5 Time processing rate obtained from analyzed strategies. It was computed taking as reference at SMOTE: $ratio_k = TT_k/TT_{SMOTE}$, where TT_k and TT_{SMOTE} are the training time (measured in minutes) of the strategy k and SMOTE, respectively.

Dataset	MSEBPOS+GGE	SMOTE+GGE[1]	MSEBPOS	Imbalanced[1]	SMOTE[1]
MCAA	0.20	0.63	0.22	0.40	1.00
MCAB	0.26	0.79	0.30	0.48	1.00
MCAC	0.29	0.74	0.37	0.61	1.00
MFEA	0.31	0.69	0.37	0.53	1.00
MFEB	0.27	0.87	0.40	0.56	1.00
MFEC	0.38	0.82	0.52	0.50	1.00
MSAA	0.21	0.45	0.49	0.61	1.00
MSAB	0.13	0.27	0.34	0.31	1.00
MSAC	0.12	0.15	0.36	0.32	1.00
MSEA	0.34	0.79	0.45	0.60	1.00
MSEB	0.22	0.63	0.28	0.40	1.00
MSEC	0.29	0.73	0.50	0.58	1.00
M92A	0.41	0.96	0.32	0.44	1.00
M92B	0.35	0.99	0.35	0.33	1.00
M92C	0.53	0.99	0.48	0.25	1.00
Average	0.29	0.70	0.38	0.46	1.00
Average Rank	1.27	3.60	2.40	2.73	5.00

[1] Classification using SBP

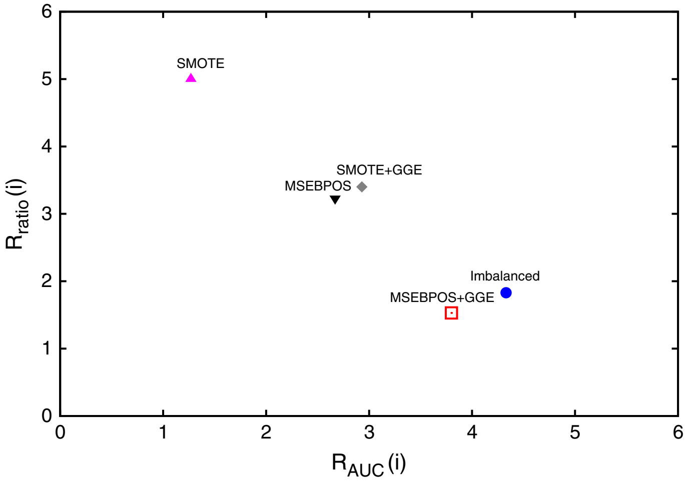

Fig. 2 Set size ratio versus AUC plot using average ranks

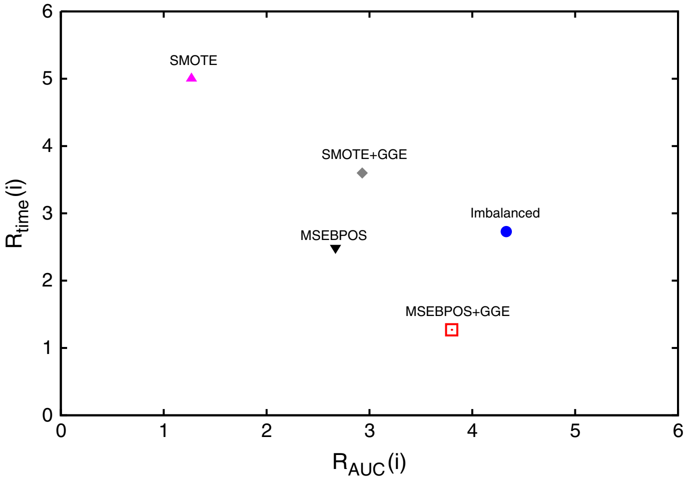

Fig. 3 Time processing rate versus AUC plot using average ranks

appears as the second algorithm with a good trade-off. In the case of SMOTE and imbalanced approaches, these lies the furthest from the origin (0, 0).

6 Conclusions

This paper has proposed a MSE back-propagation over-sampling technique for learning multi-class imbalance data sets. The method has been proposed based upon the SBP algorithm. The aim of this alternative is to identify a suitable over-sampling rate, whilst reducing the processing time and storage requirements, as well as, keeping or increasing the performance of predictive models.

Experimental results over fifteen high imbalanced multi-class data sets have demonstrated that the MSEBPOS algorithm achieve competent results in terms of the AUC measure, processing time and storage requirements with respect to the original SMOTE technique. Also, an analysis with Bonferroni-Dunn post-hoc model test has allowed to observe that MSEBPOS behaves similarly to this one. When visualizing the AUC results and the time processing rate, as well as, the set size ratio, we have found that the strategies based on the MSEBPOS, yield the most balanced trade-off between the three rates.

Future work will extend this study in order to find the new mechanics to identify the most appropriate over-sampling ratio which allows to improve significantly the classification performance of the proposed method but keeping its advantage in terms of NN training time. On other hand, would be interesting for future work to generalize the proposed method, i.e, to balance the training dataset through of an effective over-sampling technique and to use the MSE to automatically identify the optimal amount of samples as the minorities as the majorities classes so the resulting method may be considered a under and over sampling technique. In addition, we will want to dip in the analysis of multi-class learning problems where the dataset shows a large size and an extreme class imbalance.

Acknowledgments This work has partially been supported by the Mexican SEP under grants PROMEP-/103.5/11/3796 and PROMEP/103.5/12/4783, the TESJo under grant SDMAIA-010, and the Mexican Science and Technology Council (CONACYT-Mexico) through a Postdoctoral Fellowship [223351].

References

1. A. Asuncion, D.N.: UCI machine learning repository (2007). www.ics.uci.edu/mlearn/
2. Alejo R, Valdovinos RM, García V, Pacheco-Sanchez JH (2012) A hybrid method to face class overlap and class imbalance on neural networks and multi-class scenarios. Pattern Recognit Lett 34(4):380–388
3. Anand R, Mehrotra K, Mohan C, Ranka S (1993) An improved algorithm for neural network classification of imbalanced training sets. IEEE Trans Neural Netw 4:962–969
4. Batista GEAPA, Prati RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explor Newsl 6:20–29
5. Batista GEAPA, Prati RC, Monard MC (2005) Balancing strategies and class overlapping. In: IDA, pp. 24–35
6. Bruzzone L, Serpico S (1997) Classification of imbalanced remote-sensing data by neural networks. Pattern Recognit Lett 18:1323–1328
7. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
8. Chawla NV, Cieslak DA, Hall LO, Joshi A (2008) Automatically countering imbalance and its empirical relationship to cost. Data Min Knowl Discov 17:225–252
9. Crone SF, Lessmann S, Stahlbock R (2006) The impact of preprocessing on data mining: an evaluation of classifier sensitivity in direct marketing. Eur J Oper Res 173(3):781–800
10. Debowski B, Areibi S, Gréwal G, Tempelman J (2012). A dynamic sampling framework for multi-class imbalanced data. ICMLA 2:113–118
11. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7(1):1–30
12. Fawcett T (2006) An introduction to roc analysis. Pattern Recogn Lett 27:861–874
13. Fernández-Navarro F, Hervás-Martínez C, Antonio Gutiérrez P (2011) A dynamic over-sampling procedure based on sensitivity for multi-class problems. Pattern Recogn 44(8):1821–1833
14. Fernández-Navarro F, Hervás-Martínez C, García-Alonso CR, Torres-Jiménez M (2011) Determination of relative agrarian technical efficiency by a dynamic over-sampling procedure guided by minimum sensitivity. Expert Syst Appl 38(10):12483–12490
15. García S, Herrera F (2009) Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy. Evol Comput 17:275–306
16. García V, Sánchez JS, Mollineda RA (2008) On the use of surrounding neighbors for synthetic over-sampling of the minority class. In: Proceedings of the 8th conference on Simulation., modelling and optimization, SMO'08Stevens Point, Wisconsin, USA, pp 389–394

17. Han H, Wang W, Mao B (2005) Borderline-smote: a new over-sampling method in imbalanced data sets learning. ICIC 1:878–887
18. Hand DJ, Till RJ (2001) A simple generalisation of the area under the roc curve for multiple class classification problems. Mach Learn 45(2):171–186
19. Haykin S (1999) Neural networks. A comprehensive foundation, 2nd edn. Pretince Hall, New Jersey
20. He H, Bai Y, Garcia EA, Li S (2008) Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: IJCNN, pp. 1322–1328
21. He H, Garcia E (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21(9):1263–1284
22. Iman RL, Davenport JM (1980) Approximations of the critical region of the friedman statistic. Commun Stat Theory Methods 9(6):571–595
23. Japkowicz N, Stephen S (2002) The class imbalance problem: a systematic study. Intell Data Anal 6(5):429–449
24. Kotsiantis SB (2007) Supervised machine learning: a review of classification techniques. In: Emerging artificial intelligence applications in computer engineering, pp. 3–24
25. Kretzschmar R, Karayiannis NB, Eggimann F (2005) Feedforward neural network models for handling class overlap and class imbalance. Int J Neural Syst 15(5):323–338
26. Lawrence S, Burns I, Back A, Tsoi A, Giles CL (1998) Neural network classification and unequal prior class probabilities. In: Neural networks: tricks of the trade, LNCS. pp 299–314
27. Lecun Y, Bottou L, Orr GB, Müller KR (1998) Efficient backProp. In: G. Orr, K. Müller (eds.) Neural networks-tricks of the trade, lecture notes in computer science, vol. 1524, pp. 5–50. Springer Verlag
28. Li BY, Peng J, Chen YQ, Jin YQ (2006) Classifying unbalanced pattern groups by training neural network. ISNN 2:8–13
29. Moscato P, Cotta C (2003) A gentle introduction to memetic algorithms. Handbook of metaheuristics, international series in operations research and management science. Springer, New York, p 105144
30. Murphey YL, Guo H, Feldkamp LA (2004) Neural learning from unbalanced data. Appl Intell 21(2):117–128
31. Oh SH (2011) Error back-propagation algorithm for classification of imbalanced data. Neurocomputing 74(6):1058–1061
32. Orriols-Puig A, Bernadó-Mansilla E, Goldberg DE, Sastry K, Lanzi PL (2009) Facetwise analysis of xcs for problems with class imbalances. Trans Evol Comp 13:1093–1119
33. Ou G, Murphey YL (2007) Multi-class pattern classification using neural networks. Pattern Recognit 40(1):4–18
34. Provost F (2000) Machine learning from imbalanced data sets 101. In: Proceedings of the learning from imbalanced data sets: Papers from the Amercian association for artificial intelligence workshop, 2000 (Technical report WS-00-05)
35. Ramanan S, Clarkson T, Taylor J (1998) Adaptive algorithm for training pram neural networks on unbalanced data sets. Electron Lett 34(13):1335–1336
36. Wang S, Yao X (2012) Multiclass imbalance problems: analysis and potential solutions. IEEE Trans Syst Man Cybern Part B 42(4):1119–1130
37. Weiss GM, Provost FJ (2003) Learning when training data are costly: the effect of class distribution on tree induction. J Artif Intell Res 19:315–354
38. Wilamowski BM, Kaynak O (2001) An algorithm for fast convergence in training neural networks. In: Proceedings of the international joint conference on neural networks, 2:17781782
39. Zhou ZH, Liu XY (2006) Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Trans Knowl and Data Eng 18:63–77